SIMULATION OF AN EVENT BASED RAINFALL-RUNOFF PROCESS : AN ARTIFICIAL NEURAL NETWORK APPROACH

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by

S.K.V. PRASAD INDURTHY

to the

DEPARTMENT OF CIVIL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
JULY, 1999

CERTIFICATE

It is certified that the work contained in the thesis entitled "**SIMULATION OF AN EVENT BASED RAINFALL – RUNOFF PROCESS : AN ARTIFICIAL NEURAL NETWORK APPROACH** ", by **S.K.V.Prasad Indurthy** (Roll No. 9720312), has been carried out under my supervision and that this work has not been submitted elseware for a degree.

(Dr. Ashu Jain)
Assistante Professor
Department of Civil Engineering
Indian Institute of Technology
Kanpur, INDIA

July 30, 1999

Acknowledgements

I take this opportunity to express my deep sense of gratitude to Dr. Ashu Jain for the constant inspiration, support and encouragement given to me, not only during the course of this thesis, but also at other critical times of my M.Tech program. His exemplary guidance, commendable patience benefited me immensely in completing this work. Special thanks are due to him for advising me to develop a professional outlook towards whatever I do, be it report writing or meeting a deadline. It was indeed a great privilege for me to have worked under his supervision.

I thank gratefully Prof. T. Gangadharaiah for his prudent guidance and clear deep insight into problems during my M.Tech program. I am also thankful to Dr. Bithin Dutta and Dr. Rajesh Srivastava for their kind cooperation during my M.Tech program.

I consider myself fortunate to have nice friends at IITK. In fact it was their support, inspiration and love that made my stay memorable and delightful here. To list them is a Herculean task. Special thanks are due to Kishore, Ramesh babu and Waheed for rendering their help in thesis work. I acknowledge the help received from Laxmi narayana, Pakumar and Ravi in doing this thesis.

I will be accused of injustice if I do not appreciate the help and support rendered by my batch mates Siva, Manohar, Sridhar, Vijay mishra, Ramakrishna, Swaroop and Jeenu. I enjoyed partying, discussing any problem with them. I enjoyed a lot with with my juniors. In their company I always relaxed from the tension of work and enjoyed most. Their concern, love and support during crisis are some of the things I will never forget.

These words and space are not real tribute to the blessings and countless sacrifices of my parents. Whatever I am today it is their encouragement and unseen prayer of silence at God for me. My elder brother Bujji annayya willingly taken

up tasks to make me feel comfortable. His inspiring presence has been instrument in uplifting my confidence. My younger brothers Malli and Kiran, sister Kali akka also stand by me in completing my M.Tech program.

July 30,1999

S.K.V.Prasad Indurthy
Indian Institute of Technology
Kanpur

Abstract

Three types of modeling approaches have been used to simulate an event based rainfall-runoff process. The data, consisting of four storms, of the Salado Creek tributary at Bitters Road, San ANTONIO, Texas were employed in this study. The first type of models were Artificial Neural Network models, the second type of models are regression models, and the third type of model uses Unit hydrograph approach. Specifically, three ANN models, six regression models including linear, polynomial and non-linear models, and two Unit Hydrograph models were developed. For all the models developed in this study data from three storms were used for calibration purposes whereas the data from the fourth storm were used to test the performance of the models using certain statistical parameters. It has been found in this study that the ANN model provide a better representation of the rainfall-runoff process as compared to the other coventional models investigated in this study.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General

The effective and efficient operation of a watershed system requires an accurate assessment of the short term forecasts of the stream flows. The short term stream flow forecasts can be obtained by developing some type of mathematical model for the Rainfall-Runoff (R-R) process. Models are simplified systems that are used to represent real life systems and are substitutes of the real systems for certain purposes. The models express formalized concepts of the real systems. Development of mathematical models relating the rainfall incident upon a catchment to the stream flow emanating from it has received a great deal of attention from researchers and hydrologists working in this area. R-R modeling requires complex analysis involving temporal and/or spatial variations of rainfall and hydrologic abstraction such as infiltration, evaporation, transpiration etc., and as such is best suited for use with digital computers. Both the amount of effort and the complexity of models seem to have increased continually with the expansion in available computing power.

A catchment model is a set of mathematical expressions describing relevant phases of the hydrologic cycle, with the objective of simulating the conversion of rainfall into runoff. Generally, the values of various parameters in a rainfall-runoff model must be selected so that runoff calculated from the model matches recorded

runoff from same historical period. While the physics governing the path of a drop of water through a catchment to the stream involves complex relationships, the information contained in rainfall-runoff observations is sufficient to support models of only very limited complexity. Consequently, the quality or the performance of a model is a function of the quality of the data employed.

1.2 Models for Rainfall-Runoff Process

Models of rainfall-runoff process can be grouped into two types, namely, deterministic models and stochastic models. A deterministic model is formulated by following physical processes as described by differential equations. A deterministic model is formulated in terms of a set of variables and parameters and equations relating to them. Deterministic models of rainfall-runoff process can be one of two kinds; namely, physically based models, or conceptual models. In physically based models, we use equations of mass, energy and momentum to describe the movement of water over the earth surface, and through the saturated and unsaturated zones of the earth. Conceptual model is a simplified representation of the physical processes, obtained by lumping spatial and/or temporal variations, and described in terms of ordinary differential equations. The conceptual models of rainfall-runoff process may further be classified in to two classes; namely, event based models and continuous models. Event based models are short term, designed to simulate individual rainfall-runoff events. On the other hand, a continuous model allows simulation of stream flow for time intervals greater than the storm duration.

Conceptual models ignore the spatially distributed, time varying and stochastic properties of Rainfall-Runoff process, they attempt to incorporate realistic representation of the major non-linearities inherent in the Rainfall-Runoff process. Implementation and calibration of a conceptual model is difficult requiring sophisticated mathematical tools. Most of the conceptual models are lumped representation of the 'parameters'.

Stochastic model are formulated by following laws of chance or probability. A

stochastic model identifies a relationship between input; such as, antecedent flows, and rainfall; and output; such as, stream flow, without attempting to describe any of the internal mechanism of the process of transformation. Stochastic models are also known as black-box models or system theoretic models. A Black-Box model uses an appropriate mathematical function or functions which are fitted to the data without considering the processes it represents. Black-Box models are easy to develop and implement. They normally take less time and effort.

More recently, Artificial neural networks (ANNs) have been successfully used as tools for modeling and forecasting. ANNs are relatively black-box or system theoretic type models based on the neural structure of the brain. The brain basically learns from experience. ANNs can be regarded, in one respect, as multivariate nonlinear analytical tools, and are known to be very good at recognizing patterns from noisy, complex data, and estimating their nonlinear relationships. Many studies have shown that ANNs have the capability to learn the underlying mechanics a physical process. Based on some of its successful applications, it is evident that the ANN technology can be applied to many real world problems, especially in engineering.

Regardless of the kind of the model, the essential ingredients of a model are variables and parameters. Variables are the physical quantities themselves i.e., discharge, flow area, rainfall and so on. Parameters are the quantities that control the behavior of the variables. A catchment model is referred to as either deterministic, conceptual or empirical, depending on whether the majority of its components and parameters have a deterministic, conceptual or empirical basis.

A typical modeling application consists of the following steps (i) Selection of model the type (ii) Model Formation (iii) Parameter Estimation (iv) Model Testing and (v) Model Application.

The first step in the model development is the selection of model type i.e., which type of model is to be developed e.g., deterministic, conceptual or stochastic and whether lumped or distributed etc. Model formulation involves deciding

on the structure of the model to be developed. This may involve deciding the variables involved. Model calibration or parameter estimation is the process by which the values of model parameters are identified for use in a particular application. It consists of the use of the rainfall-runoff data and procedure to identify the model parameters that provide the best agreement between simulated and recorded flows.

To evaluate the predictive accuracy of a model, it is customary to divide the data available into two distinct sets (1) Calibration set and (2) Verification set. The first set is used in the calibration whereas the second set is used in a model verification i.e., to measure of accuracy of the calibration. Once the model has been calibrated and the parameters verified, it is ready to be used in the predictive stage of modeling.

1.3 Objectives of the Present Study

The primary objective of this study is to investigate the new technique of ANN for use in modeling the rainfall-runoff process. Because of the data availability, an event based rainfall-runoff process model will be developed using ANN technique. It is hoped that this study will demonstrate the applicability of the system theoretic ANN approach in developing effective non linear models of rainfall-runoff process of a catchment. The process of developing an ANN model will require the following steps.

1. The first step will be to collect data in terms of both rainfall and runoff.

2. The second step will be to develop a computer code for simulating an ANN model. This ANN model to be developed will have a general architecture.

3. Once the computer code is developed some computational experiments will be conducted to check the correctness of the developed code using hypothetical and real data.

4. Once the computer code is verified, the next step will be to identify optimal

ANN structure for simulating rainfall-runoff process. This will require a trial and error procedure to come up with the best model architecture.

5. Once the ANN models have been developed this will be tested on a different set of data to verify the performance in terms of some statistical parameters.

The secondary objective of this study is to develop some deterministic models and stochastic models for rainfall-runoff process using the same data set. This is necessary to compare the performance of the developed ANN models with the conventional models.

The long term objective of this study is to contribute to the research initiative in the area of ANN application to water resources system and hydraulic engineering specially for rainfall-runoff modeling. It is hoped that this research effort will initiate a program of research which will continue and evolve over a long-period of time in the area of ANN application to water resources systems.

1.4 Organization of the Thesis

The first chapter of the thesis describes the problem of rainfall-runoff modeling in general, various modeling techniques available, and the objectives and organization of the thesis. Chapter 2 reviews the literature available in the area of rainfall-runoff modeling. Chapter 3 presents an introduction to the new technique of ANNs. The next chapter, chapter 4, discusses the development of the various types of model structures investigated in this study. Discussion of results is taken up in chapter 5 whereas chapter 6 presents the conclusions and scope for future research work in the area of rainfall-runoff modeling. References and appendices are provided at the end.

Chapter 2

Literature Review

There are many models that have been developed so far for rainfall-runoff process. These range from linear hydrograph to complex model such as Stanford Watershed Model (SWM). In this chapter, an attempt has been made to review various types of models reported in literature in brief.

The modern development in deterministic modeling of rainfall -runoff process was linear models (Sherman, 1932). The basic assumption of linear systems analysis in rainfall-runoff process was that the process is linear and time invariant. Example of linear model is the linear storage model. The theory of the instantaneous unit hydrograph (IUH) is based on the concept of a linear storage resulting from a hypothetical linear reservoir. Mathematically, it can be represented as follows:

$$I - Q = dS/dt \tag{2.1}$$

$$S = KQ \tag{2.2}$$

Where I represents inputs to the system, Q represents output from the system, S is the storage in the system and K is a constant having dimensions of time. Equation 2.1 is the equation of continuity whereas equation 2.2 represents the basic property of a linear reservoir.

re are a number of papers available on modeling of individual components und water flow, unsaturated soil water flow and channel routing, but there literature available on the deterministic models dealing with interacting es and the application to real world problems. Some examples are System ɔgique European (SHE) model (Beven *et al.*1980) and Institute of Hydrol- tributed Model (IHDM), (Morris, 1980).

C-SMA models i.e., Sacramento Soil Moisture Accounting Model, is a typ- moisture accounting model developed by Burnish et al. (1973) mainly for recasting purposes. In this model, rainfall incident upon a catchment is red as falling on one of two types of areas, previous areas and impervious th connected to the stream channels. Runoff is produced from the imper- eas by any rainfall event no matter how small. The runoff occurs from the s area only when the rainfall intensity exceeds storages. The storage rep- he volume of precipitation required to meet the basic requirements and is r closely bound to soil particles. When this tension storage is filled, water nulated in the upper zone as the free water storage from where water can deeper storage or move laterally to appear in the stream channel as inter e vertical drainage water, or percolation, can enter one of the three lower rages, the lower zone tension storage (the water held by the soil particles) two lower zone free water storages that are available for drainage as base sub surface outflow. These storages fill simultaneously but drain inde- y at different rates. The surface runoff and inter flow are routed to the nt outlet by a non-dimensional unit hydrograph.

npts to calibrate SAC-SMA model for river and flood forecasting, have ble to obtain unique optimal parameter estimates using automatic calibra- edures (Johnston and Pilgrim1976; Pickup 1977; Sorooshian 1978; La- 981; Sorooshian and Gupta, 1983; Hendrickson *et al.*, 1988). One of the ptimization method, the shuffled complex evolution (SCE-UA) method o find the optimal parameter set during calibration of the SAC-SMA ian, Duan, and Gupta, 1993). In brief, the SCE-UA method involves the

initial selection of a "population" of points distributed randomly throughout the feasible parameter space. The population is partitioned into several "complexes". Each complex is then allowed to "evolve" so as to independently search the parameter space in a manner that is based on an extension of the simplex local-search algorithm. After a prescribed number of steps, the complexes are "shuffled" together and new complexes formed such that the information gained separately by each complex is shared. The evolution and shuffling procedures are repeated until prescribed stopping criteria are satisfied. The SAC-SMA model has not gained much popularity and has been of interest to the researchers and scientists only. This may be because of its complex structure and the data requirements.

Another computer model for Rainfall-Runoff modeling that is used most widely is HEC-1. HEC -1 was developed by U.S.Army Corps of Engineers, at the Hydrologic Engineering Center (Hydrologic Engineering centre, 1981). In this model, stream channel routing is accomplished by standard hydrologic methods. The model does not account for the dynamic effects that are present in river of mild slope. The main purpose of HEC-1 is to simulate hydrologic processes during flood events. Therefore, it has no provision for soil moisture recovery during periods of no precipitation with simulations being limited to a single-storm event. The process of converting precipitation to direct runoff can be simulated by HEC-1 for small sub basins or large complex watersheds. In addition to R-R modeling HEC-1 has the following capabilities: (i) Optimal estimation of unit hydrograph, loss rate, and stream flow routing parameters from measured data (ii) Comparison of damage frequency curves and expected annual damages for various locations and multiple flood control plans and (iii) Simulation of reservoir outflow for dam safety analysis.

The most comprehensive rainfall-runoff model available on computer is the Stanford Watershed Model (SWM)(Linsley and Crawford 1960). The SWM produces estimates of daily flows from daily rainfall inputs using infiltration, unit hydrograph and recession functions. This was further modified by the inclusion of soil moisture budgeting, evapotranspiration estimates and flow routing techniques to give hourly interval estimates. In SWM calculations begin from known

or assumed initial conditions and are continued until the time sereis input data are exhausted. Precipitation is stored in three soil-moisture storages. The three soil moisture storages are (i) upper zone storage, (ii) lower zone storage and (iii) ground water storage. The upper and lower zone storage account for overland flow, infiltration, interflow and inflow to groundwater storage. The upper zone simulates hydrological abstractions and runoff resulting from minor (frequent) storms, including the first few hours of major (infrequent) storms. Conversely, the lower zone accounts for the hydrologic abstractions and runoff during major storms. Groundwater storage supplies baseflow to stream channels. Evaporation and transpiration may occur from any of these storages. The runoff from overland flow, interflow and baseflow enters the channel system and is routed down stream to the watershed outlet, where it is expressed as a contineous outflow hydrograph (Crawford and Linsley 1966). Continuous development of this model has resulted in the hydrologic simulation program (Johanson *et al.*, 1980). A parallel development due to Sugawara known as the tank model, simulated the movement of water through the system using simple linear reservoirs arranged in series and in parallel (Sugawara 1981).

A lot of work has been reported in the literature in the area of rainfall-runoff modeling using black-box or system theoretic models. Some notable examples include linear time series models such as ARMAX i.e., Auto Regressive Moving Average with Exogenous Inputs. Some notable examples include Jacoby, 1966; O'Connel, 1974; Box and Jenkins, 1976; Treiber and plate, 1977; Bras, Kottegoda and Horder, 1980; Pergram, 1980; Salas *et al.*, 1980; Chang *et al.*, 1982; Rodriguez-Iturbe, 1985.

Recently, new technique of ANNs has been proposed as a tool for modeling in forecasting. An ANN is a flexible mathematical structure which is capable of representing complex non-linear process that simulate the inputs and outputs of any physical system. Rainfall-Runoff modeling using ANN approach is reported in literature. Some of the papers using ANN for research papers using ANN for R-R modeling are discussed here.

Kuo-lin Hsu, *et al.* (1993) presented a new procedure for identifying the structure and parameters of three-layer feed forward ANN models and demonstrated the potential of such models for simulating the nonlinear hydrologic behavior of watersheds. A new procedure for estimating the weights (parameters of a three-layer ANN) was described as well. They compared the performance of that model to that of of an optimal ARMAX model identified using classical time series methods and to the SAC-SMA model used by the U. S. National Weather Service. The non-linear ANN model approach is shown to provide a better representation of the rainfall-runoff relationship of the medium-size Leaf River basin near Collins, Mississippi, than the linear ARMAX time series approach or the conceptual SAC-SMA model.

The inversion of snow water equivalent, SWE, from passive remote microwave sensing measurements may be accomplished by using a neural network trained with a dense media multiple scattering model [Chang and Tsang 1992]. Mason developed ANN models using radial basis function effectively to simulate rainfall-runoff process with huge amount of data. This model is based on the assumption that runoff depends on time, rainfall intensity I, the rate of change of I and the integral of I [Mason 1997]. Some other successful applications of ANN in R-R modeling include Zhang [1998]; Kuo-lin Hsu, Hoshin [1998]; Cho-Chung Yang [1998].

It is obvious from some of the work on R-R modeling using ANNs that the ANNs have great potential that needs to be explored. This research work makes an effort in this direction.

Chapter 3

Artificial Neural Networks

3.1 General

Artificial Neural networks (ANNs) are biologically inspired structures that can be used for modeling physical systems. They are composed of elements that perform in a manner that is analogous to the most elementary functions of human brain. ANNs exhibit a surprising number of characteristics similar to those exhibited by the human brain. For example, they learn from experience, generalize from previous examples to new ones and abstract essential characteristics from inputs containing irrelevant data. ANNs can modify their behavior in response to their surrounding environment. They self adjust to produce consistent responses (Wasserman 1989). In recent times, ANNs have been proposed as efficient tools for modeling and forecasting. However, before we go into the details of ANNs, let us look at the biological neural network of the human brain.

3.2 Biological Neural Network

The human brain is made up of vast network of small processing units called "neurons" or "neurodes". The average human brain, roughly 1.4 kg in weight and 1500 cubic centimeters in volume, is estimated to contain about 100 billion cells of various types. A neurons is a special cell that conducts an electrical signal, and there are about 10 billion neurons in the human brain. Each neuron is about

one-hundredth size of the period at the end of a sentence.

A typical neuron has three parts: the cell body or soma, the axons, and the dendrites. The structure of a biological neuron is shown in Figure 3.1. Dendrites form a dendritic tree, which is a very fine bush of thin fibers around the neurons body. Dendrites receive information from neurons through axons which are long fibers that serve as transmission lines. An axon is a long cylindrical connection that carries impulses from the neuron. The end portion of an axon splits into a small end-bulb almost touching the dendrites of neighboring neurons. The connection between axon of one neuron and dendrite of the other is called a "Synapse".

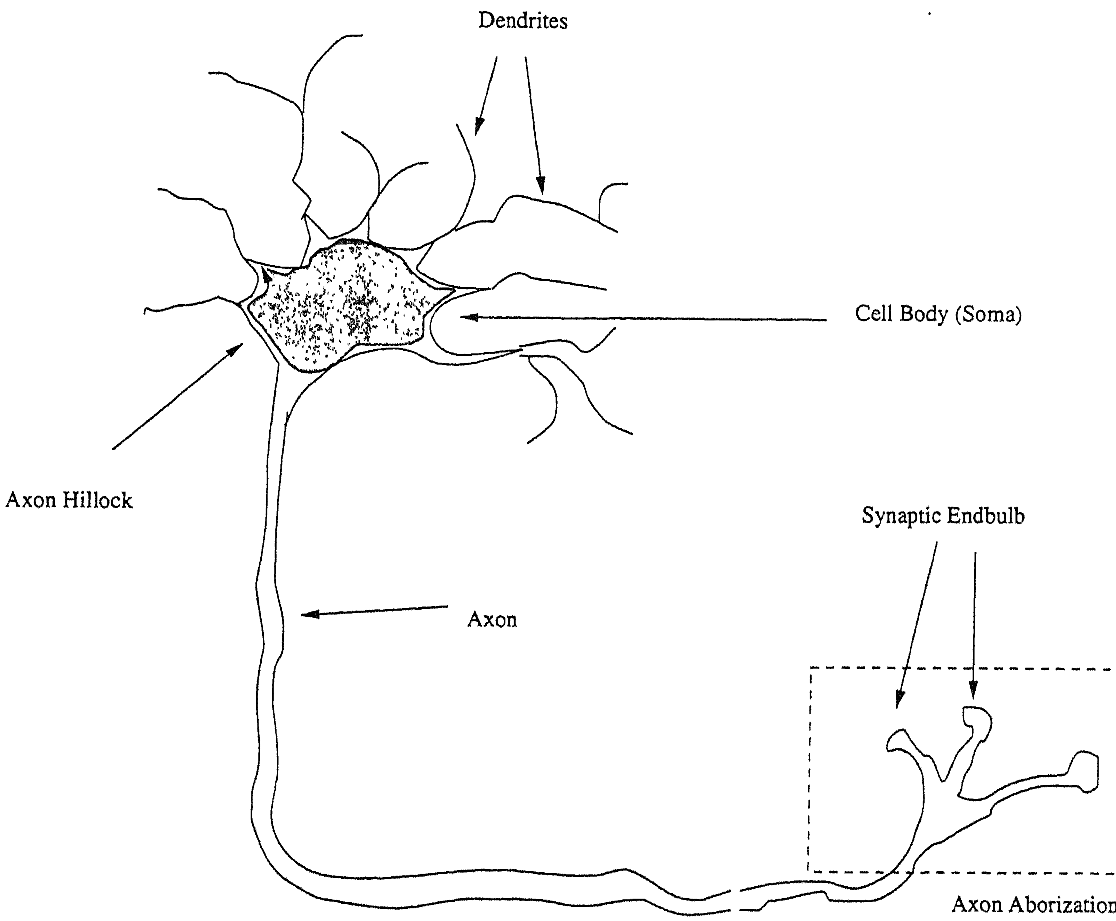


Figure 3.1: The Structure of a Biological Neuron

The human nervous system may be viewed as a three-stage system as depicted

in the block diagram shown in Figure 3.2 (Arbib, 1987). Central to the system is the *brain*, represented by the *neural (nerve) net* in this figure, which continually receives information, perceives it, and makes appropriate decisions. The *receptors* convert stimuli from the human body or the external environment into electrical impulses that convey information to the neural net (brain). The *effectors*, on the other hand, convert electrical impulses generated by the neural net into discernible responses as system outputs.

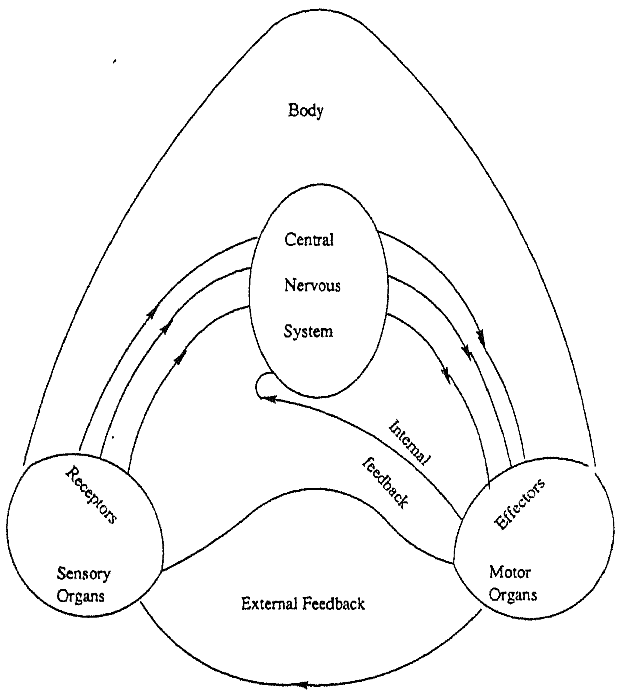


Figure 3.2: Information Flow in Human Nervous System

The brain is a highly *complex, nonlinear, and parallel computer* (information-processing system). It has the capability of organizing neurons so as to perform certain computations (e.g., pattern recognition and perception) many times faster than the fastest digital computer in existence today. Consider, for example, human *vision*, which is an information-processing task (Churchland and Sejnowski, 1992; Levine, 1985; Marr, 1982). It is the function of the visual system to provide a *representation* of the environment around us, and more importantly, to supply the information we need to *interact* with the environment. To be specific, the brain routinely accomplishes perceptual recognition tasks (e.g., recognizing fa-

miliar face embedded in an unfamiliar scene) in something of the order of 100-200 milliseconds, whereas tasks of much lesser complexity will take days on a huge conventional computer (Churchland, 1986). It is important to recognize that the structural levels of organization in a biological neural network described herein are a unique characteristic of the brain. They are nowhere to be found in a digital computer, and we are nowhere close to realizing them with artificial neural networks. The artificial neurons used to build ANNs are truly primitive in comparison to those found in the human brain.

3.3 Artificial Neural Networks (ANNs)

An Artificial Neural Network (ANN) is an analogy to the brain. Work on ANN, commonly referred to as "neural networks", has been motivated right from its inception by the recognition that the brain computes in an entirely different way from the conventional digital computer. In its most general form, an ANN is a machine that is designated to *model* the way in which the brain performs a particular task or function of interest; the network is usually implemented using electronic components or simulated in software on a digital computer. To achieve good performance, ANNs employ a massive interconnection of simple computing cells referred to as "neurons" or "processing units".

ANNs derives its computing power through, first, its massively parallel distributed structure and, second, its ability to learn and therefore generalize; generalization refers to the neural network producing reasonable outputs from inputs not encountered during training (learning). Training or learning of an ANN is a procedure in which the connection strength are updated using certain laws and principles. These two information-processing capabilities make it possible for ANNs to solve complex (large-scale) problems that may be intractable otherwise. The use of neural networks offers the following useful properties and capabilities:

Nonlinearity: A neuron is basically a nonlinear device. Consequently, a neural network, made up of an interconnection of neurons, is itself nonlinear. Moreover, the nonlinearity in an ANN is of a special kind in the sense that it is dis-

tributed throughout the network. Nonlinearity represented in the ANNs is a highly important property, particularly if the underlying physical mechanism responsible for the generation of an output signal (e.g. runoff) is inherently nonlinear.

***Input-Output Mapping*:** A popular paradigm of learning called *supervised learning* involves the modification of the weights (each connection strength in the ANN is expressed by a numerical value called a weight) of an ANN by applying a set of labeled *training samples* or *task examples*. Each example consists of a unique *input signal* and the corresponding *desired response*. The weights (free parameters) of the network are modified so as to minimize the difference between the desired response and the actual response of the network produced by the input signal until an appropriate statistical criterion is satisfied. The training of the network is repeated for many examples in the set until the network reaches a steady state, where there are no further significant changes in the weights; the previously applied training examples may be re applied during the training session but in a different order. Thus the network learns from the examples by constructing an *input-output mapping* for the problem at hand.

***Adaptivity*:** Neural networks have a built-in capability to *adapt* their weights to changes in the surrounding environment. In particular, a neural network trained to operate in a specific environment can be easily *retained* to deal with minor changes in the operating environmental conditions. Moreover, when it is operating in a *non stationary* environment (i.e. on whose statistics change with time), a neural network can be designed to change its weights in real time. The natural architecture of an ANN coupled with its adaptive capability, make it an ideal tool for use in developing models for on line applications such as drought and flood related applications.

3.4 ANN Architecture

The manner in which the neurons of an ANN are structured is intimately linked with the learning algorithm used to train the network. The architecture of an ANN can be one of the following two types i.e. single-layered or multi-layered

architectures. These are discussed in brief next.

3.4.1 Single-Layer ANNs

A *layered* neural network is a network of neurons organized in the form of layers. In the simplest form of a layered network, we just have an *input layer* of source nodes that projects onto an *output layer* of neurons (computation nodes), but not vice versa. In other words, this network is strictly of a *feed forward* type. It is illustrated in the following figure for the case of three nodes in both the input and output layers. Such a network is called a *single-layer network*, with the designation "single layer" referring to the output layer of computation nodes (neurons). In other words, we do not count the input layer of source nodes, because no computation is performed there. However, if combinations are performed at the input layer also, then this layer may also counted. In that case, this ANN may be referred to as a double-layer ANN.

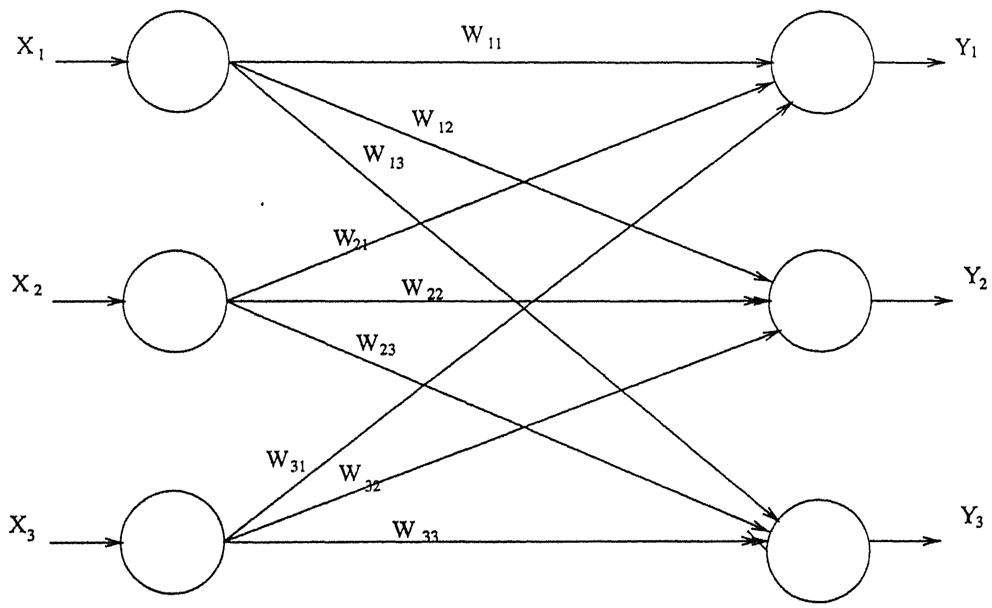


Figure 3.3: Single Layer Artificial Neural Networks

As shown in the Figure 3.3, X_1, X_2 and X_3 are inputs. Circles are the neurons or nodes. Each neuron simply computes output of a weighted sum of the inputs to the network. The connection between the neurons, represented by lines, is

quantified by their weights which are shown in the form of w_{ij}. Y_1, Y_2 and Y_3 are the output from the single-layer ANN.

3.4.2 Multi Layer ANNs

Larger, more complex networks generally offer greater computational capabilities. Although networks have been constructed in every imaginable configuration, arranging neurons in layers mimics the layered structure of certain portions of the brain. These multilayer networks have been proven to have capabilities beyond those of a single layer.

A multilayer ANN with two hidden layer network with one input layer and one output layer is shown in Figure 3.4. Inputs are shown by X1, X2, X3,.... and Wkl is the weight matrix from layer l to layer k. Circles represent the neurons and 'O' is the observed output of the network. The most commonly employed method of training a multi-layered ANN by scientists and engineers is the back propagation algorithm which is desribed next.

3.4.2.1 Back Propagation Algorithm

Back propagation algorithm, classified as supervised learning, is a systematic method of training multilayer ANNs. A typical back propagation neural network will consist of an input and output layer of neurons plus one or more hidden layers of neurons. The back-propagation training through generalized delta rule of learning is an iterative gradient descent algorithm designed to minimize the mean-square error between the actual output of a multi-layer feed forward ANN and the desired output. It requires continuous differentiable non-linear function in the ANN to compute output from each neuron. Summary of the Error Back-Propagation Training Algorithm is given here.

The input x_i to any neuron i is given by

$$x_i = \sum_{i=1}^{N} O_j W_{ji} \tag{3.1}$$

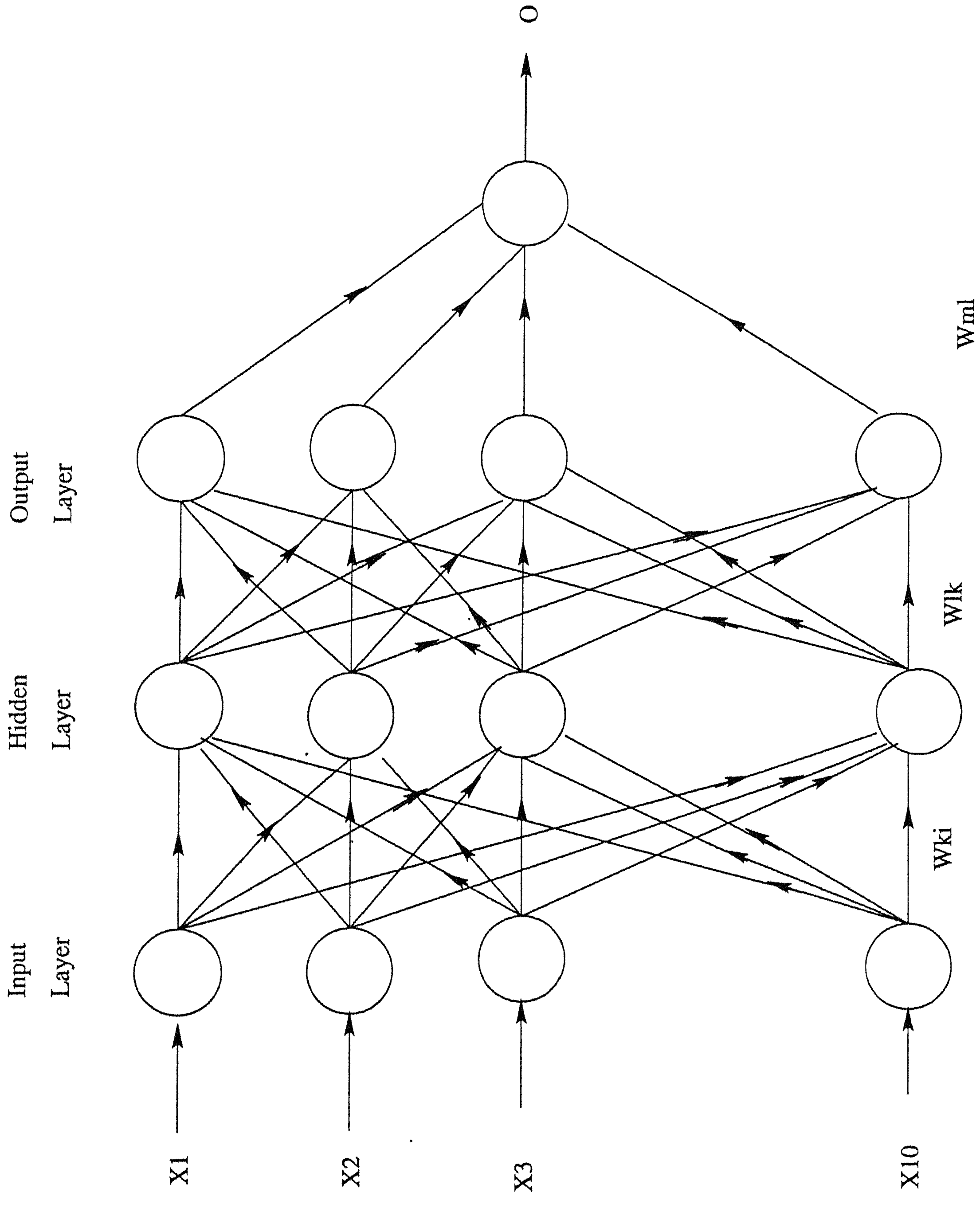


Figure 3.4: Multilayer Artificial Neural Network

where Wji = weight matrix from i th layer to the jth layer.

N = Total number of inputs

Based on the magnitude of input x_i, each neuron i fires or generates an output signal O_i:

$$O_i = \frac{1}{1 + exp(-x_i a)} \tag{3.2}$$

where a is an integer positive number.

The weighted sum of all these outputs acts as the input to the neurons of subsequent layers.

The difference between the computed and desired output is called the error:

$$e_i = (O_i - D_i) \tag{3.3}$$

$$E = \sum_{i=1}^{N} e_i^2 \tag{3.3a}$$

where O_i = Observed output from the ith neuron in the output layer of the ANN.

D = Desired output at the ith neuron in the output layer of the ANN.

Before starting the training of an ANN all the weights are randomized to some initial value. In the learning mode, the ANN considers each data set from the training patterns one at a time and generates output. Then it compares its output with the target output of all the nodes of output layer and if there is any discrepancy, the error is back propagated by changing the interconnection weights according the following equations:

$$W_{ji}(n+1) = W_{ji}(n) + \Delta W_{ji}(n) \tag{3.3b}$$

$$\Delta W_{ji}(n) = \beta(\delta_i)O_j + \alpha(\Delta W_{ji}(n-1)) \tag{3.4}$$

where $\Delta W_{ji}(n)$ is the change in weight w_{ji} at nth iteration.

$\Delta W_{ji}(n-1)$ is the change in weight w_{ji} at (n-1)th iteration.

$W_{ji}(n+1)$ is the updated value of weight W_{ji} at (n+1)th iteration

$W_{ji}(n)$ is the updated value of weight W_{ji} at (n)th iteration

O_j is the output from jth neuron in the output layer.

β is learning constant

α is momentum constant

For output neuron the value of δ_i is given by:

$$\delta_i = O_i(1 - O_i)(D_i - O_i) \qquad (3.5)$$

Where O_i is the output from the network

D_i is desired value of the output

But for hidden layers target output is unknown. So the weighted sum of the error signals of all the neurons of the succeeding layers is utilized to calculate error signal of any neuron *i* in the hidden layer. The expression is as follows:

$$\delta_i = O_i(1 - O_i)\sum_m \delta_m W_{ml} \qquad (3.6)$$

where m runs over all the neurons in the subsequent layer. This value of δ_i is then substituted in the equation 3.4. This procedure or updating the coefficients W_{ji} is repeated until the errors are minimized up to the acceptable levels or tht convergence is achieved.

3.5 Activation Function

The activation function, denoted by f(x), defines the output from a neuron in terms of the activity level as its input. Many activation functions have been used e.g.,threshold function, sigmoid function, ramping function, hyperbolic tangent function etc. Activation function most commonly employed by engineers is the sigmoid function.

The sigmoid function is defined as a strictly increasing function that exhibits smoothness and asymptotic properties. The sigmoid function can be given by the following equation:

$$f(x) = \frac{1}{1 + exp(-ax)} \tag{3.7}$$

where a is the *slope parameter* of the sigmoid function.

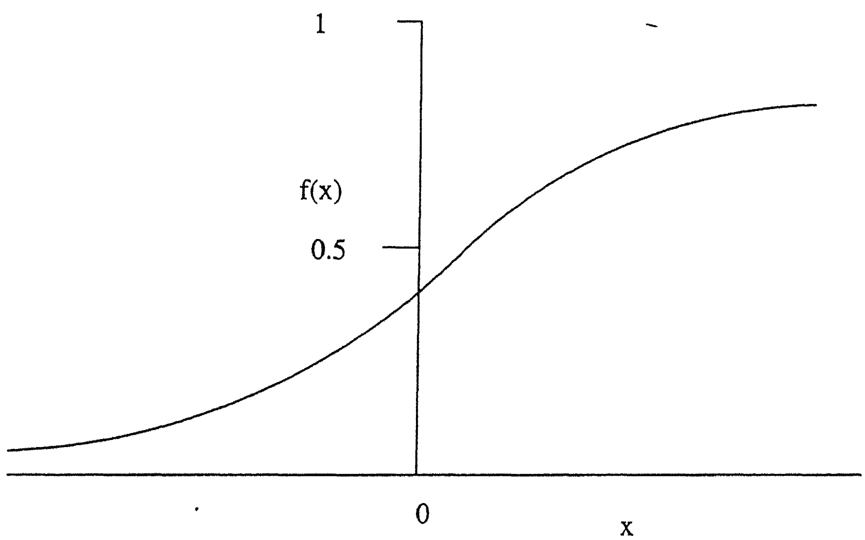


Figure 3.5: Sigmoid Function

By varying the parameter a, we obtain sigmoid functions of different slopes. Some of the property that make the sigmoid function suitable for modeling are that (a) it is bounded above and below, (b) it is monotonically increasing, (c) it is continuous and differentiable and (d) its first derivative can be represented in terms of the function itself.

3.6 Initial Weights

The weights of an ANN to be updated/optimized are typically initialized at small random values. The initialization strongly affects the ultimate solution. If all

weights start out with equal values, and if the solution requires that unequal weights be developed, the ANN may not train properly. Unless the ANN is disturbed by random factors or the random character of input patterns during training, the internal representation may continuously result in symmetric weights.

Also, the ANN may fail to learn from the set of training examples with the error stabilizing or even increasing as the learning continues. In fact, many empirical studies on the training algorithms point out that continuing training beyond a certain low-error plateau results in the undesirable drift of weights. This causes the error to increase and the quality of mapping implemented by the network decreases. To counteract the drift problem, network learning should be restarted with other random weights (Zurada 1997). Normal guidelines suggest that all weights be initialized in the ranges ± 0.3, ± 0.5 or ± 0.7 depending upon the particular application. The choice of initial weights is, however, only one of several factors affecting the training of the network toward an acceptable minimum error.

3.7 Learning Constant

The effectiveness and convergence of the error back-propagation learning algorithm depends significantly on the value of the learning constant β. In general, the optimum value of β depends on the problem being solved, and there is no single learning constant value suitable for different training cases. This problem seems to be common for all gradient-based optimization schemes.

Although the choice of the learning constant depends strongly on the class of the learning problem and on the network architecture, the values ranging from 10^{-3} to 10 have been reported throughout the technical literature as successful for many computational back-propagation experiments. For large learning constants, the learning speed can be drastically increased; however, the learning may not be exact, with tendencies to overshoot, or it may never stabilize at any minimum. Even though the simple gradient descent can be efficient, there are situations when moving the weights within a single learning step along the negative gradient vector by a fixed proportion will yield a minor reduction of error. For flat error surfaces

for instance, too many steps may be required to compensate for the small gradient value.

When broad minima yield small gradient values, then a larger value of β will result in a more rapid convergence. However, for problems with steep and narrow minima, a small value of β must be chosen to avoid overshooting the solution. This leads to the conclusion that β should indeed be chosen using a trial and error procedure for each problem. One should also remember that only small learning constants guarantee a true gradient descent. The price of this guarantee is an increased total number of learning steps that need to be made to reach the satisfactory solution. It is also desirable to monitor the progress of learning so that β can be increased at appropriate stages of training to speed up the minimum seeking.

3.8 Momentum Method

The purpose of the momentum method is to accelerate the convergence of the error back-propagation learning algorithm. The method involves supplementing the current weight adjustments with a fraction of the most recent weight adjustment. This is analogous to the moving average term in the time-series models. This is usually done according to the formula

$$\triangle W_{ji}(n+1) = \beta(\delta_i)O_j + \alpha \triangle W_{ji}(n) \tag{3.8}$$

where the arguments *n+1* and *n* are used to indicate the current and the most recent training step respectively, and α is a user-selected positive momentum constant. The second term indicating a scaled most recent adjustment of weights, is called the *momentum term*. Typically, α is chosen between 0 and 1.

3.9 Applications of ANNs in Engineering

Recently, there has been a surge in the application of ANNs to engineering. ANNs have been used in a wide variety of engineering application. In addition to some ANN applications in rainfall-runoff modeling described in chapter 2, some other examples include application of ANNs in structural analysis (Rehak, Thewalt and Doo 1989); daily electrical load forecasting (Park *et al.*1991); ground water management (Garret, Panjithan and Eheart 1992); time sereis forecasting (Sharda and Patil 1990); hourly water demand prediction (Crommelynck *et al.* 1992).

Chapter 4

Model Development

4.1 Introduction

Three types of model structures have been developed in this study. The first type of models are ANN models, the second type of models are regression models, and third type of models use the unit hydrograph approach.

Data Employed in this study were taken from U.S. Geological Survey Report, (USGS 1977). Data include storm rainfall-runoff record for Salado Creek tributary at Bitters Road, San Antonio, Texas for 1976 water year. The data consists of rainfall-runoff record for four different storm events in 1976. The first storm occurred on April 4, the second storm occurred on May 6-7 , the third storm occurred on June 16 and the final storm occurred on Aug 18, 1976. The rainfall and runoff values were observed at 5 minutes interval. The data for these storms are presented in Appendix A.

4.2 Development of the ANN Model

The development of an ANN model may be accomplished as explained in the following steps.

1. The first step is to process the available data. This requires identification of variable that will represent input and output neurodes and normalization of

data between the values 0.1 and 0.9.

2. The second step is to select an appropriate ANN architecture, with an input layer consisting of source nodes equal in number to the process variables under study/consideration. A subset of examples is then used to train the network by means of a suitable algorithm. This phase of the network design is called *learning* or training of the network.

3. The third step is to evaluate, the performance of the trained ANN with a different set of data that it has never seen before. The performance of the ANN is assessed by comparing the output reported by the network with the actual value of the output in question. This second phase of the network operation is called testing the network.

A computer program in 'C' has been developed to simulate back propagation ANN for use in rainfall-runoff modeling in this study. The flow chart for the computer program for simulating a back propagation ANN is shown in Figure 4.1.

Figure 4.1 illustrates the algorithm of the modified error back-propagation training for a basic two-layer network. The learning begins with the feed forward recall phase (Step 2). After a single pattern vector is submitted at the input layer, the layer responses are computed. Then, the error signal computation phase (Step 4) follows. Note that the error signal vector must be determined in the output layer first, and then it is back propagated toward the network input nodes. The weights are adjusted within the matrix in Step 5 for neurodes in output layer. Weights for the neurons in hidden layer are adjusted in Step 6. The learning procedure stops in step 7 when all the patterns in the training set have been presented to the ANN.

4.2.1 Verification of the Computer Program

In order to check the correctness of the computer program, a few computational experiments were conducted. First, a small ANN was trained to simulate a linear fit to the data generated using the equation of a line i.e., $y = mx + c$. To further verify the program, the data in terms of altitude and temperature published by the

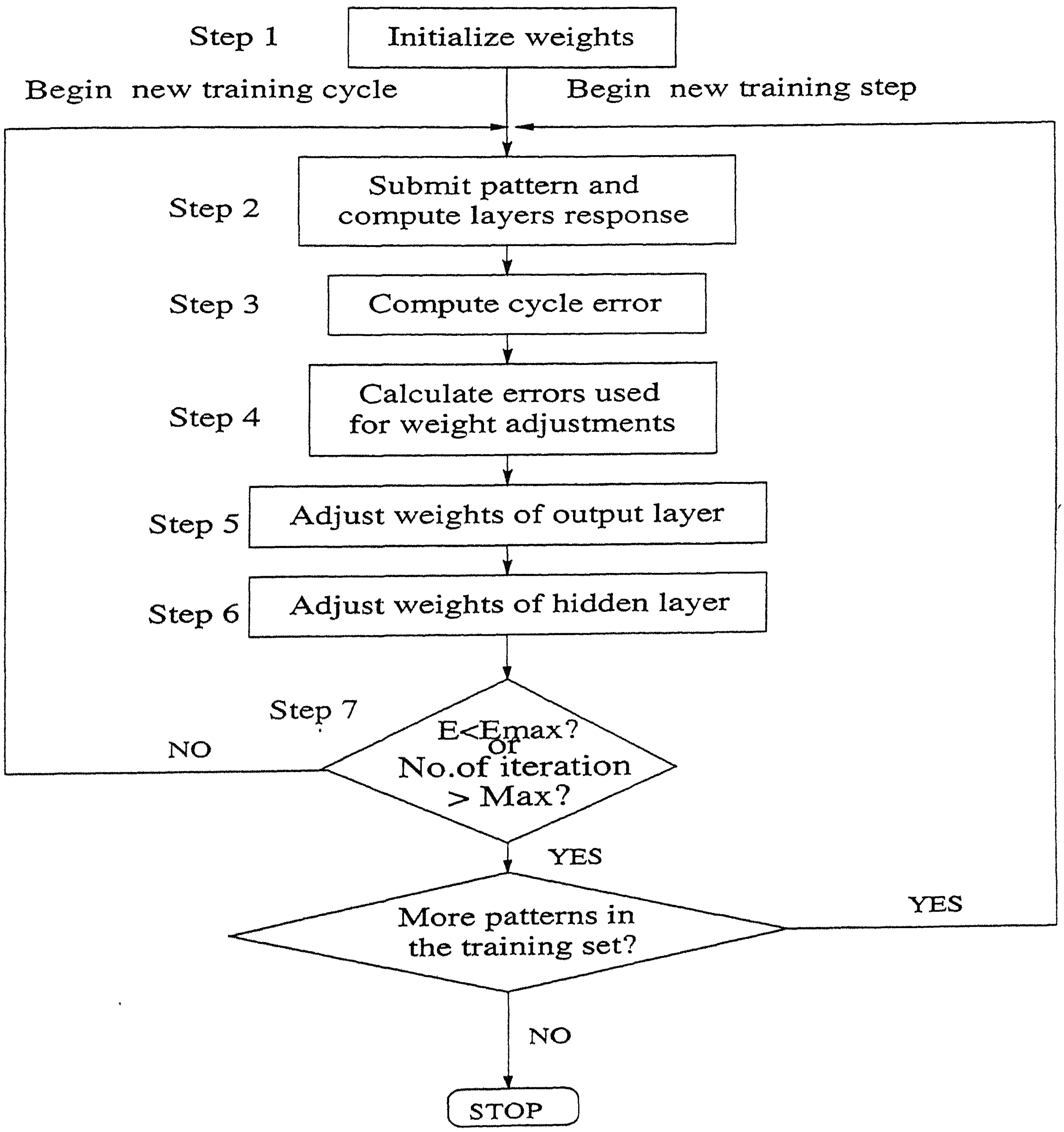


Figure 4.1: Flow Chart of Developed Computer Program

National Aeronautic and Space Administration, USA (NASA 1966)for the year 1966 were employed. Figure 4.2 shows the optimum ANN model developed for simulating altitude and temperature. Here input I represents altitude and output O represents temperature. ANN structure is 1-3-3-1 with 2 additional neurons connecting input and output. Wji is weight matrix from the i^{th} layer to the j^{th} layer.

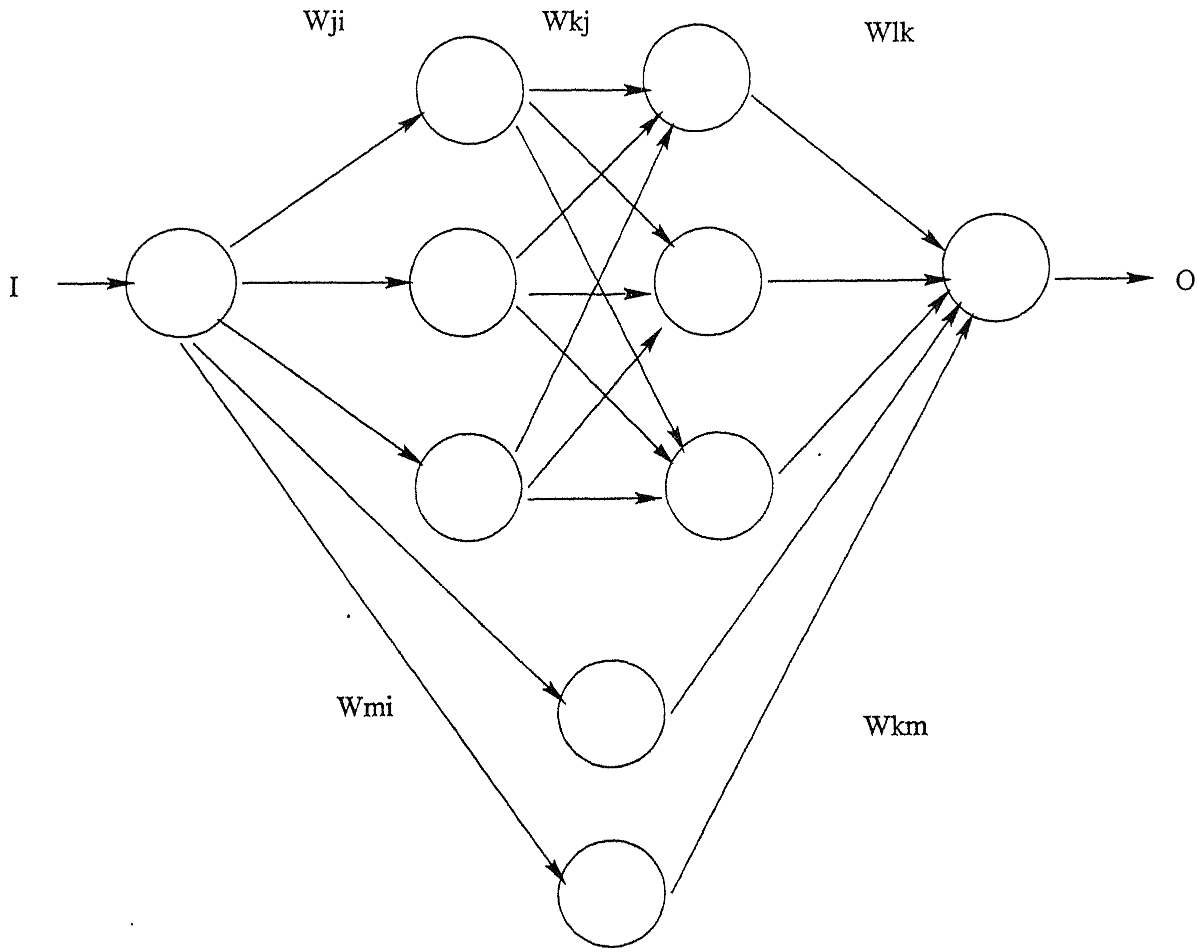


Figure 4.2: ANN Model for Simulating Altitude and Temperature Data

The statistical performance of the above ANN model is summarized in Appendix B for NASA Data. The average error from the trained network in testing 158 data was 5.14%. Figure showing the observed and modeled temperatures is also shown in Appendix B.

4.2.2 ANN Model for Rainfall-Runoff Process

The computer program developed for error back propagation training algorithm is capable of simulating any number of neurons in any of the layers of a three-layer ANN i.e. it can simulate an ANN architecture of n-k-l-m where, n is the number of neurons in input layer, l is the number of neurons in first hidden layer, k is the number of neurons in the second hidden layer, m is the number of neurons in the output layer, and m is the number of neurons in output layer.

This computer program was used to investigate various ANN architectures to simulate the R-R process for the Salado creek in San Antonio, Texas. The number of neurons in the hidden layers were determined using the principle of parsimony i.e., network with smaller number of neurons in hidden layers giving reasonable performance will be preferred over the network with larger number of neurons in the hidden layer giving similar performance.

Two types of back-propagation ANN models were developed for simulating the R-R process. The first model structures was a simple two layer ANN with one hidden layer, whereas, the second model structure was a more complex three layer ANN with two hidden layers.

4.2.2.1 Three-layer ANN model :

First, a two layer ANN model with structure 4-N-1 was developed using the data at two different locations on the salado creek having different drainage areas. This was done to develop a general model for simulating R-R process for Salado creek, San Antonio, Texas. In this model, the input layer had four neurons which represented rainfalls at various time steps and drainage area. The output layer had one neurode representing runoff at time step 't'. The number of neurons in the hidden layer were varied from 2 to 7. The results in terms of some statistical parameters from this model are presented in Appendix C. The model which gave best results was 4-6-1.

Because of the fact that 4-N-1 model intended to model R-R process in the Salado Creek watershed as a whole from two different stations. It was realized that, first a model using data from only one station should be developed. Further, It was observed that the time lag between peak rainfall and peak runoff data at Bitters Road was 35 minutes. So, the runoff output R_t was assumed to be related to past rainfall inputs $P_{(t-j)}$, in which j ranges from 1 to 8, means 8 time steps in the past. As a result, an ANN model having rainfall atleast 8 time steps in the past as input neurodes will be more appropriate. So an ANN model structure in the form of 9-N-1, was investigated where N was varied over the range 2 to 16. For this the results presented in Appendix-D. Model 9-14-1 performed the best among all structures investigated. It's performance in terms of statistical parameters is presented in next chapter. This ANN model structure was further modified by taking Runoff $R_{(t-1)}$ at previous time step as one of the neurodes in input layer giving rise to another ANN model structure 10-N-1. Results from 10-N-1 ANN models in terms of some statistical parameters are presented in Appendix D. Out of the models investigated, model 10-20-1 gave the best results. The performance in terms of some statistical parameters, for both training and testing phase are presented in the next chapter.

It can be observed that the two-layer models with more neurodes in input-layer gave reasonably good results, however, it was felt that a better performance may be achieved with a four-layer ANN model, as highly complex and non-linear relationships such as R-R process can be modeled in better way using a three-layer ANN model.

4.2.2.2 Four-Layer ANN Model

The model structure represented by the notation ANN (N, L, K, M), was developed for identifying the non-linear relationship between rainfall and runoff where 'N' is the number of neurons in the input layer, L and K are the number of neurodes in the two hidden layer and M is the number of neurodes in the output layer. The structure of the input layer for four layer model was identical to the three-layer models i.e. input neurode represnted rainfall values 8 time steps in the past. This

model was named 9-L-K-1 ANN model. The results from 9-L-K-1 ANN model in terms of some statistical parameters are presented in Appendix-E. Model 9-10-10-1 performed the best among all structures investigated. This ANN model structure was further modified by taking Runoff $R_{(t-1)}$ at previous time step as one of the neurodes in input layer giving rise to another ANN model structure i.e. 10-L-K-1. The results in terms of various statistical parameters from 10-L-K-1 ANN model are presented in Appendix F. Out of the models investigated, model 10-12-14-1 gave the best results. The performance of the 10-12-14-1 model in terms of some statistical parameters, for both training and testing, are presented in the next chapter.

After all the trial and error procedures Optimum ANN structure obtained for the present problem is given in the table given below:

Table 4.1: Optimum ANN structure of the Three Layer ANN Model

Parameter	Symbol	values
Number of neurons in input layer	N	10
Number of neurons in 1st hidden layer	L	12
Number of neurons in 2nd hidden layer	K	14
Number of neurons in the output layer	M	1
Learning constant	β	0.9
Momentum constant	α	0.7

It was observed that the error reduced during training with the number of iterations at different rates. During the initial stages of learning the error reduced drastically and it flattend slowly. It is apparent that as the error surface sets flattened the ANN stops learning. In present study it was observed that the training of the ANN can be stopped at about $50,000$ iterations since there is no significant reduction in the error after $50,000$ iterations.

To observe the performance of the ANN after it has been trained, the same data set used for training can be used for prediction also. How well the trained ANN is recognizing the patterns which it has already seen before can be judged from the results. In this study, It was observed that the ANN has memorized the

245 sets remarkably well, which means the modification of weights proceeded in the right direction. Then data were divided in two sets, one is for training (208 data points comprising of three storms) and other is for testing (37 data points for testing corresponding to the fourth storm of Aug 18). The results in terms of some statistical parameters both for training and testing sets are presented in the next chapter, from all of the ANN models.

4.3 Multiple Regression models

In this section, regression model development is described. Various regression model structures were investigated for predicting runoff which can broadly be grouped into two types i.e. linear multiple regression models and non-linear regression models. These are described next.

4.3.1 Linear Multiple Regression Model (LMRM)

A linear multiple regression model can be used for simulating rainfall and runoff. The structure of the linear multiple regression model developed in this study is presented by the following equation.

$$\mathrm{R}_t = \beta_0 + \beta_1 \mathrm{P}_t + \beta_2 \mathrm{P}_{t-1} + \beta_3 \mathrm{P}_{t-2} + \beta_4 \mathrm{P}_{t-3} + \ldots\ldots + \beta_9 \mathrm{P}_{t-8} + \beta_{10} \mathrm{R}_{t-1}$$

where β's represent regression coefficients to be determined
R's represent runoff
P's represent rainfall and
t represents time

To find the parameters of the linear regression model, NAG routines were used. For this purpose, first a file was prepared consisting of all the data in many columns. The number of columns should be equal to number of variables where each column corresponds to the observations made on a single variable. Number of rows are the total number of observations made on any variable. The last column/variable is reserved for dependent variable (R_t) only, which is a runoff at a time step 't' in this case. After the input file is prepared, NAG subroutine

G02BAF.F was used to perform the regression analysis and the coefficients of regression were obtained. The data for first three storms were used to obtain the parameters of the model. The regression coefficients for the linear multiple regression model are presented in Table 4.2.

Table 4.2: Regression Coefficients for Linear Multiple Regression Model

Coefficient	Value
β_0	0.521827
β_1	0236682
β_2	-0.033368
β_3	0.065800
β_4	0.159492
β_5	0.090896
β_6	0.007326
β_7	-0.019592
β_8	-0.046623
β_9	0.012202
β_{10}	0.14074

This model, named LMRM, was then used to predict the fourth storm and tested using various statistical parameters. The results of this testing exercise are presented in the next chapter.

4.3.2 Non Linear Regression Models

Five types of non-linear regression models structures were investigated. Out of these five models, first two are polynomial models involving only one variable whereas remaining three models are non-linear mutiple regression models involving many variables. Many polynomials were investigated ranging from second order to seventh order. All polynomial model structures used rainfall at time 't' as the independent variable and runoff at time 't' as the dependent variable. However, polynomial models of order three and six gave reasonable performance and are presented here.

4.3.2.1 Cubic Polynomial Model (POLY3)

The cubic polynomial equation employed to relate the runoff to rainfall is given below:

$$R_t = \beta_0 + \beta_1 P_t + \beta_2 P_t^{\,2} + \beta_3 P_t^{\,3} \qquad (4.2)$$

where R_t is the Runoff at time 't'
P_t is the rainfall at time 't' and
β_0,β_1, β_2, β_3 are model parameters to be determined

The model parameters were found using the NAG subroutine and are presented in the Table 4.3 below.

Table 4.3: Model Parameters for POLY3 Model

Coefficients	Value
β_0	0.064404
β_1	0.647097
β_2	-1.07663
β_3	0.592325

This model, named POLY3, was then used to predict the fourth storm and tested using various statistical parameters. The results of this testing exercise are presented in the next chapter.

4.3.2.2 Polynomial Model of Sixth order (POLY6) :

The sixth order polynomial equation employed to relate the runoff to rainfall is given below:

$$R_t = \beta_0 + \beta_1 P_t + \beta_2 P_t^{\,2} + \beta_3 P_t^{\,3} + \beta_4 P_t^4 + \beta_5 P_t^5 + \beta_6 P_t^6 \qquad (4.3)$$

The model parameters were found using the NAG subroutine and are presented in the Table 4.4 below.

Table 4.4: Model Parameters for the POLY6 Model

Coefficient	Value
β_0	-0.54690
β_1	14.12088
β_2	-135.139
β_3	645.427
β_4	-1585.15
β_5	1908.724
β_6	-886.86

This model, named POLY6, was then used to predict the fourth storm and tested using various statistical parameters. The results of this testing exercise are presented in the next chapter.

4.3.2.3 First Non Linear Multiple Regression Model (NLMRM-1)

This model had the following structure.

$$R_i = \beta_0 + \beta_1 P_t^2 + \beta_2 P_{t-1}^2 + \dots\dots\dots\dots + \beta_9 P_{t-8}^2 + \beta_{10} R_{t-1}^2 \tag{4.4}$$

Regression coefficients obtained using NAG subroutine are presented in the given Table 4.5. This model, named NLMRM-1, was then used to predict the fourth storm and tested using various statistical parameters. The results of this testing exercise are presented in the next chapter.

4.3.2.4 Second Non Linear Multiple Regression Model (NLMRM-2)

This model had the following structure.

$$R_i = \beta_0 + \beta_1 P_t^3 + \beta_2 P_{t-1}^3 + \dots\dots\dots\dots\dots\dots + \beta_9 P_{t-8}^3 + \beta_{10} P_{t-1}^3 \tag{4.5}$$

Coefficients obtained using NAG subroutine are presented in the Table 4.5

This model, named NLMRM-2, was then used to predict the fourth storm and tested using various statistical parameters. The results of this testing exercise are presented in the next chapter.

4.3.2.5 Non Linear Multiple Regression Model (NLMRM-3)

This model had the following structure.

$$R_i = \beta_0 + \beta_1 P_t^2 + \beta_2 P_{t-1}^2 + \beta_3 P_{t-2}^2 + \beta_4 P_{t-3}^2 + \beta_5 P_{t-4}^3 + \beta_6 P_{t-5}^3 + \beta_7 P_{t-6}^3 + \beta_8 P_{t-7}^3 + \beta_9 P_{t-8}^4 + \beta_{10} P_{t-1}^4 \quad (4.6)$$

Coefficients obtained using NAG subroutine are presented in the Table 4.5. This model, named NLMRM-3, was then used to predict the fourth storm and tested using various statistical parameters. The results of this testing exercise are presented in the next chapter.

Table 4.5: Regression Coefficients of Various Non-Linear Multiple Regression Models

Parameters	NLMRM-1	NLMRM-2	NLMRM-3
β_0	0.10641	0.15068	0.14492
β_1	0.30132	0.45186	0.21803
β_2	0.14066	0.39589	0.23673
β_3	0.46792	0.591460	0.52534
β_4	0.29655	0.57682	0.63169
β_5	0.14385	0.49884	0.51967
β_6	0.20861	-0.17024	0.00527
β_7	0.06140	0.16109	0.20571
β_8	0.02356	0.32454	0.49381
β_9	0.31530	0.53629	0.67642
β_{10}	0.92790	1.10624	1.07417

4.4 Unit Hydrograph Models

Two different Linear Systems models were developed using Unit Hydrograph (UH) approach. For this purpose, one can first develop UH using observed rainfall-runoff data and then use the UH to calculate the runoff hydrograph for a given storm. In this study, two storms, namely April 4 storm and June 16 storm, were used to derive UHs. The duration of rainfall for April 4th storm was 2-hours whereas that for June 16th storm was 1-hour. Thus the storms of April 4th storm

and June 16th storm were used to develop UHs of 2-hour and 1-hour durations respectively. The UHs were derived using rainfall and runoff data and calculating the ϕ indices. The value of ϕ index for April 4th storm is 0.5192 in/hr and that for June 16th storm is 0.10587 in/hr. These UHs were then converted into 1/2 hour UHs using S-curve technique. The 1/2 hour UH derived from April 4th storm and June 16th storm are presented in Table 4.6.

Once the 1/2 hour UHs were developed, they were used to predict the Aug 18th storm. The statistical parameters quantifying errors in prediction were calculated. The results in terms of statistical parameters from the two UH models are presented in the next chapter.

Table 4.6: U.H. Ordinates of April 4 and June 16 Storms in Cu.ft/sec

Time	April 4 storm	June 16 Storm
0	0	0
5	0	0
10	0	52.80
15	31.28	75.90
20	59.72	75.90
25	201.93	135.31
30	315.70	333.33
35	310.01	336.63
40	267.35	293.72
45	213.31	234.32
50	170.65	184.81
55	142.21	148.51
60	125.14	118.81
65	110.92	69.91
70	96.70	0
75	53.80	0
80	0	0

Chapter 5

Results and Discussions

5.1 General

In this study, three types of models have been developed for modeling an event based rainfall-runoff process. The first type of models developed are ANN models, the second type of models are regression models and the third type of models use the unit hydrograph method. For modeling R-R process, rainfall-runoff data for 1976 water year at Salado creek tributary at Bitters Road, San Antonio, Texas were employed. The data consists of four storms of which three storms (April 4, May 6 and June 16) were used for model training, while the remaining storm (Aug 18) was used for evaluating the performance of various model structures developed in this study.

5.2 Statistical Parameters

Performance of each model is evaluated in terms of six different types of statistical parameters. Each statistical parameters quatifies the errors from the models or the performance of the model. These statistical parameters are described below.

5.2.1 Average Absolute Relative Error :

Average absolute relative error (AARE) is the average of the absolute value of the error in forecasting certain number of data points. To compute AARE, first the relative error in forecasting needs to be calculated.

Relative error is a measure of the error in forecasting a particular variable relative to it's exact value. Mathematically, relative error (RE) can be defined as follows.

$$RE[t] = \frac{RO(t)-RF(t)}{RO(t)} X100 \tag{5.1}$$

Where RE[t] = Relative error in forecasting RO(t).

RO(t) = Observed Runoff at time t.

RF(t) = Forecasted Runoff of at time t.

It is clear that the value of RE(t) can be either positive or negative. Once RE(t) is known, AARE can be computed as follows.

$$AARE = \frac{1}{N} \sum_{t=1}^{N} RE[t] \tag{5.2}$$

Where AARE = Average absolute relative error.

N= total number of data points forecasted.

As obvious from its definition, lower AARE values will indicate better model performance, and vice-versa.

5.2.2 Relative Error in Peak Flow

Peak flow of any hydrograph is an important parameter characterizing the behaviour or catchment response to rainfall. Relative error in peak flow is the ratio of the difference between observed peak flow of hydrograph and calculated peak flow of hydrograph from a particular model to the observed peak flow, taken as a percentage. Mathematically,

$$RE[Q_p] = \frac{DQ_p - CQ_p}{DQ_p} X100\% \tag{5.3}$$

where RE[Qp] = Relative error in peak flow
CQp = Calculated peak flow
DQp = Desired or observed peak flow

As obvious from its definition, lower value of $|RE[Q_p]|$ will indicate better model performance and vice-versa.

5.2.3 Relative Error in Peak Time

Time to peak in a hydrograph is also an important parameter characterizing the catchment response to rainfall. Relative error in peak time is the ratio of the difference between observed time to peak and the calculated time to peak from a particular model to the observed time to peak, taken as a parameter. Mathematically

$$RE[t_p] = \frac{Dt_p - Ct_p}{Dt_p} X100 \tag{5.4}$$

where $RE[t_p]$ = Relative error in peak time
Dt_p= Desired value of peak time
Ct_p = Calculated value of peak time.

As obvious from its definition, lower value of $|RE[t_p]|$ will indicate better model performanceand vice-versa.

5.2.4 Threshold Statistics

Threshold statistic, a measure of the model performance, is defined for a certain level of relative error, say x. The threshold statistic can be defined as the percentage of data points predicted for which the relative error is less than a certain level of relative error (say x%).

$$TS_{-x} = \frac{n}{N} X 100\% \qquad (5.5)$$

where n = The number of data points whose relative error is less than x.
N = Total number of data points forecasted.

As obvious from definition higher is the value of threshold statistics better is the model performance and vice-versa.

5.2.5 Correlation Coefficient

The correlation coefficient measures the correlation between calculated and observed value of the variable being modeled (runoff at a time step 't' in this case). Range of correlation coefficient can be from -1.0 to +1.0 with value close to one indicating high correlation and value close to zero indicating low correlation. Correlation coefficient can also be viewed as a measure of the performance of the model. Higher value indicates good model performance and vice-versa. Mathematically, it can be obtained using following equation:

Correlation Coefficient (R^2) = $\sum x_1 x_2 \,/\, \sqrt{\{\sum x_1^2 \sum x_2^2\}}$

$x_1 = (x_1 - \overline{x})$
$x_2 = (x_2 - \overline{x})$
where x_1 is the deviation of observed value from its mean and x_2 is the deviation of calculated value from its mean

Performance of various models in terms of various statistical parameters is presented in Tables 5.1 and 5.2. Observed and forecasted storm from various models both for trining and testing are shown in Figures 5.1 to 5.9.

Table 5.1: Statistical Performance of Models

MODEL	AARE	corr	RE[Qp]	RE[t_p]
TRAINING				
9-14-1	25.92	0.7085	6.4	5.4
10_20_1	1.409	0.9989	0	0
10_12_14_1	0.07	0.999	0	0
LMR	79.08	0.6784	1.1	4.1
NLMR1	26.37	0.8657	1	2.67
POLY3	38.22	0.3817	7.8	8.3
POLY6	68.93	0.4578	48	4.11
NLMR2	38.41	0.7979	14.4	3.4
NLMR3	44.27	0.7588	13.3	2.67
TESTING				
9-14-1	46.01	0.3702	22.72	13.78
10_20_1	18.70	0.9667	3.55	2.8
10_12_14_1	14.49	0.9630	17	0
LMR	40.31	0.9482	18.5	2.66
NLMR1	28.30	0.9488	30	2.66
POLY3	33.12	0.2559	71.4	50
POLY6	74.57	0.3169	18.8	27.7
NLMR2	48.05	0.7992	52.8	8.88
NLMR3	52.56	0.9232	57	12
UH2	101.6	0.4482	154.5	8.5
UH1	70.51	0.6810	81.8	14.2

Table 5.2: Statistical Performance of Models

MODEL	TS_0.25	TS_0.5	TS_0.75	TS_1	TS_5	TS_10	TS_25	TS_50	TS_100
TRAINING									
9-14-1	0.20	0.70	2.44	6.301	22.85	33.61	62.44	85.31	89.98
10_20_1	88.97	89.38	89.79	89.79	93.06	96.32	98.77	99.59	100.00
10_12_14_1	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
LMR	0.00	0.00	0.0000	0.0000	0.0000	9.67	9.67	38.70	70.96
NLMR1	0.40	00.81	1.63	2.44	11.42	18.77	45.30	95.51	99.18
PO3D	0.96	00.96	00.96	00.96	3.36	6.73	38.46	65.86	100.0
POLY6	0.00	0.0000	0.0000	0.0000	6.7	9.6	19.23	40.86	78.3
NLMR2	0.81	00.81	1.22	1.63	08.97	14.28	34.69	56.73	99.18
NLMR3	0.00	0.0000	0.0000	0.0000	04.08	10.61	30.61	53.06	98.77
TESTING									
9-14-1	0.00	0.00	0.00	0.00	2.70	6.45	10.44	25.37	66.42
10_20_1	0.00	0.00	0.00	0.0000	5.40	16.21	89.18	97.29	100.00
10_12_14_1	0.00	0.00	0.00	2.70	13.51	35.13	89.18	100.00	100.00
LMR	0.00	0.00	0.00	0.0000	6.45	32.25	48.38	67.74	100.00
NLMR1	0.00	0.00	0.00	0.0000	8.10	10.81	40.54	89.18	100.00
PO3D	0.00	0.00	0.00	0.0000	0.0000	10.81	51.35	75.67	100.00
POLY6	0.00	0.00	0.00	0.0000	2.77	8.35	13.88	27.77	86.17
NLMR2	0.00	0.00	0.00	2.70	2.70	10.81	21.62	45.94	94.59
NLMR3	0.00	0.00	0.00	0.0000	5.40	10.81	18.91	35.13	97.29
UH2	0.00	0.00	0.00	7.532	7.532	37.512	57.5	62.5	75
UH1	0.00	0.00	0.00	7.532	37.512	37.512	50.04	62.50	75

5.3 Discussions of Results

After examining the above results, following observations can be made.

During training the smallest AARE of 0.07% was obtained from 10-12-14-1 ANN model whereas largest AARE of 79.08% was observed from linear multiple regression model. Similarly during testing the smallest AARE of 14.49% was obtained from 10-12-14-1 ANN model whereas largest AARE of 101.6% was obtained from UH2 model. The ANN model 10-12-14-1 performed the best in terms of AARE during both training and testing. In general, the ANN models tend to have the smallest AARE during both training and testing. Non Linear Multiple regression models have the larger AARE during training when compared with polynomial models, but perform better during testing. Regression models tend to have the smaller AARE than that from the unit hydrograph models.

During training the smallest RE[Q_p] of 0% was obtained from 10-12-14-1 ANN model whereas largest RE[Q_p] of 48% was observed from polynomial model of sixth degree. Similarly during testing the smallest RE[Q_p] of 3.55% was obtained from 10-12-14-1 ANN model whereas largest RE[Q_p] of 154.5% was obtained from UH2 model. During training the smallest RE[t_p] of 0% was obtained from 10-12-14-1 ANN model and 10-20-1 model whereas largest RE[t_p] of 4.11% was observed from polynomial model of sixth degree. Similarly during testing the smallest RE[t_p] of 0% was obtained from 10-12-14-1 ANN model whereas largest RE[t_p] of 27% was observed from polynomial model of sixth degree. The 10-20-1 ANN model perfored the best in terms of RE[t_p] and RE[Q_p].

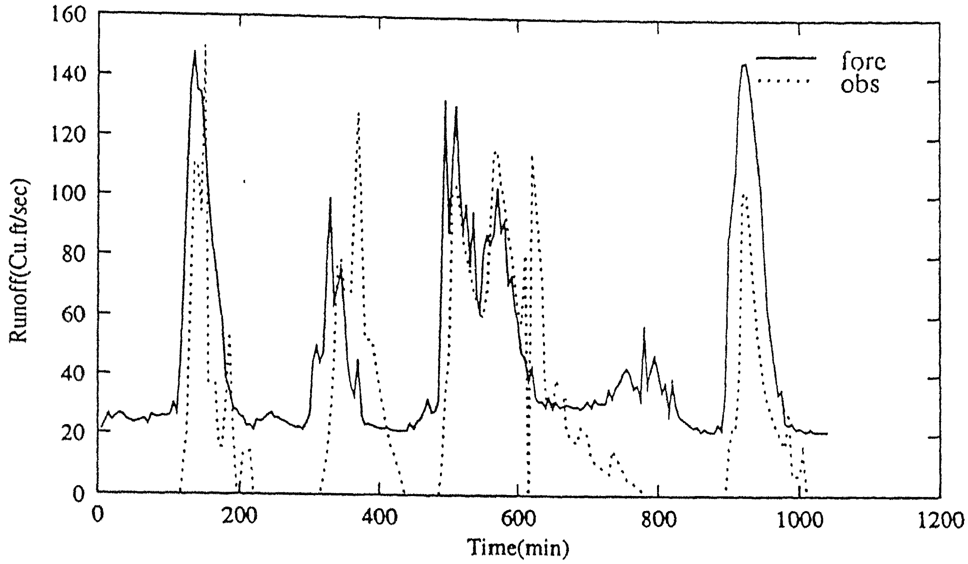
During training the larger TS_1 of 100% was obtained from 10-12-14-1 ANN model whereas smallest TS_1 of 4.11% was observed from polynomial model of sixth degree. Similarly during testing the largest TS_1 of 7.530% was obtained from UH2 model whereas smallest TS_1 of 0% was observed from four models. During training the larger TS_50 of 100% was obtained from 10-12-14-1 ANN model whereas smallest TS_50 of 40% was observed from polynomial model of sixth degree. Similarly during testing the largest TS_50 of 100% was

obtained from 10-12-14-1 ANN model whereas smallest TS_50 of 27% was observed from polynomial model of sixth degree. During training the larger TS_100 of 100% was obtained from 10-12-14-1 ANN model, 10-20-1 ANN model and Cubic Polynomial model whereas smallest TS_100 of 70.96% was observed from Linear mutiple regression model. Similarly during testing the largest TS_100 of 100% was obtained from 10-12-14-1, 10-20-1 ANN models, Linear Mutiple regression model, NLMR1 and Cubic polynomial model whereas smallest TS_100 of 75% was observed from polynomial model of sixth degree. It can be seen that the ANN model 10-12-14-1 performed the best in terms of threshold statistics also. During training 100 percent of the cases the relative error were less than 0.25% from 10-12-14-1 ANN model whereas the value of TS_0.25 from regression models ranged from 0% to 0.9%. In general, the ANN models performed best in terms of threshold statistics.

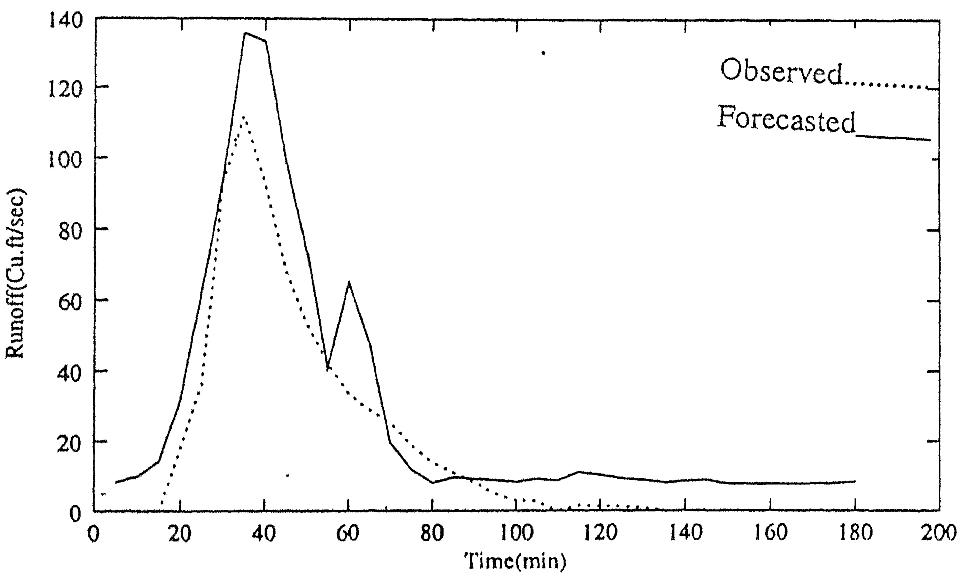
During training the larger correlation coefficient of 99.9% was obtained from 10-12-14-1 ANN model, whereas smallest correlation coefficient of 38.17% was observed from Cubic polynomial model. Similarly during testing the largest correlation coefficien 96.6% was obtained from 10-20-1 ANN model whereas smallest correlation coefficient of 25.59% was observed from Cubic polynomial model. In general, the ANN models tend to have the highest correlation coefficient both training and testing.

Observed and forecasted hydrographs from various models for training data set are shown in figures 5.1 to 5.5.The ANN models match the observed hydrograph most closely in training. So relative error in peak flow and relative error in peak time for ANN models are zero during training. But during testing ANN models tend to have small deviations from the hydrograph. Though regression models tend to have the less deviations at peak flow and peak time, but they tend to have the larger deviations in the remaining parts of the hydrograph. For polynomials, higher relative error in both peak flow and peak time were observed.

Finally, a comparison was made among the performance of the ANN models, the regression models, and the UH models. In this analysis, a particular statistic was averagedout from all models using same technique. In other words average

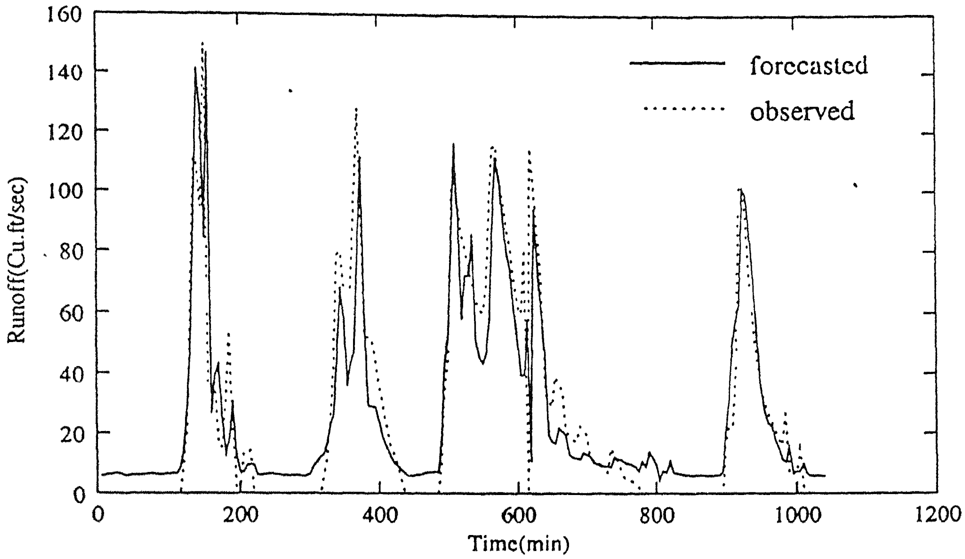


During Training

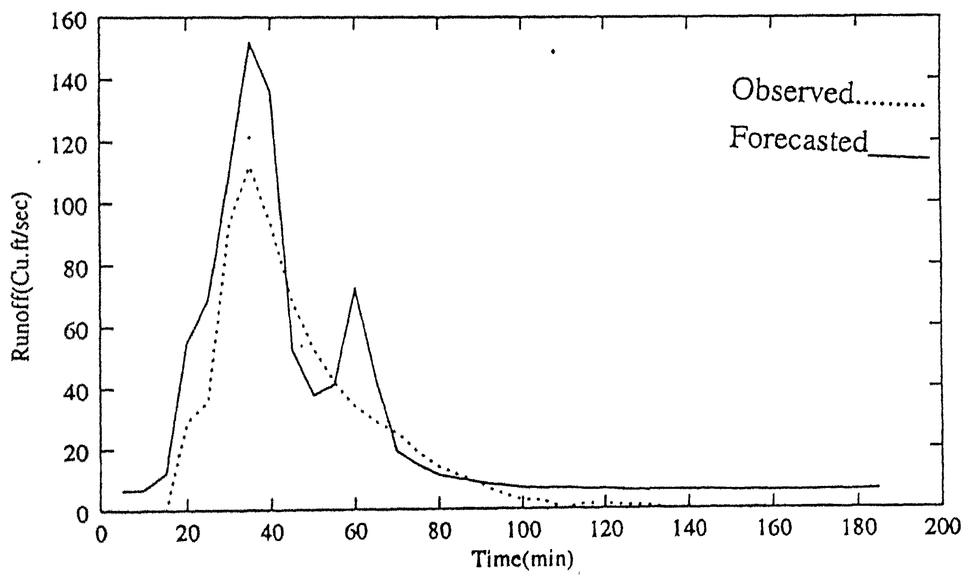


During Testing

Figure 5.1: Observed and Forecasted Runoff from LMRM

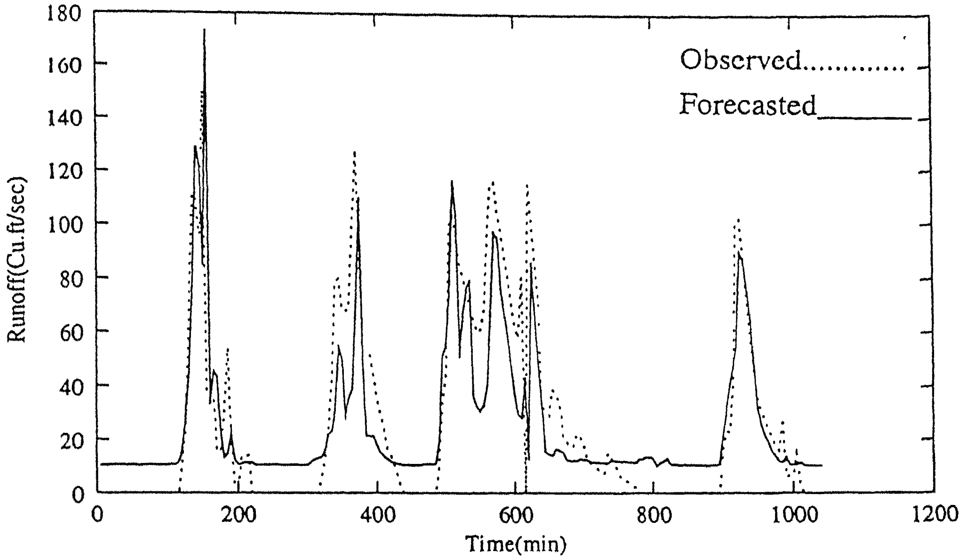


During Training

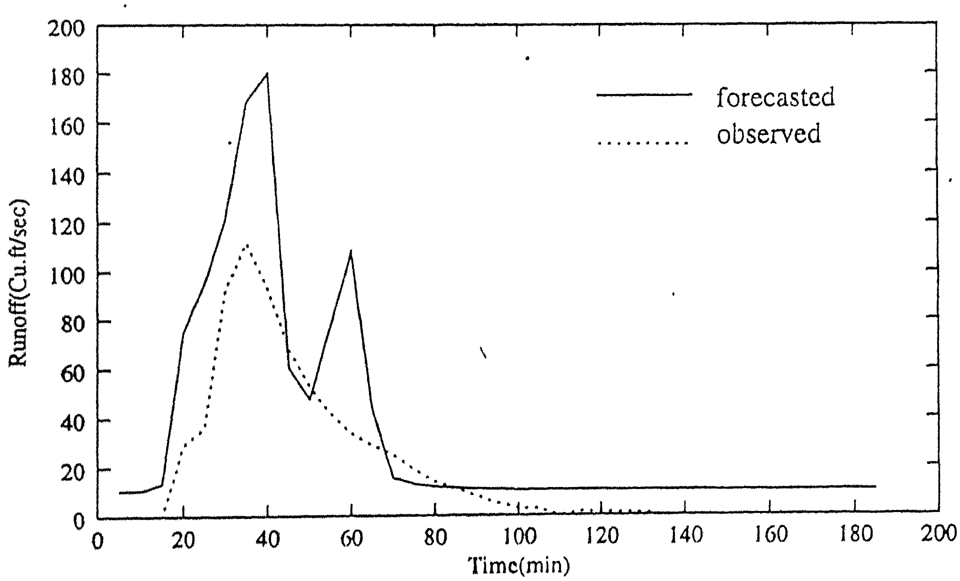


During Testing

Figure 5.2: Observed and Forecasted Runoff from NLMRM 1

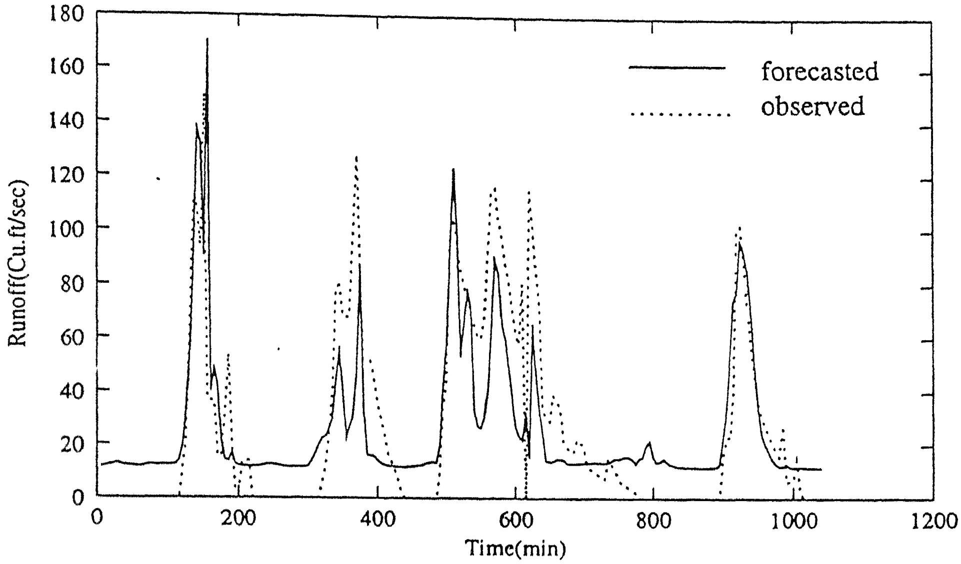


During Training

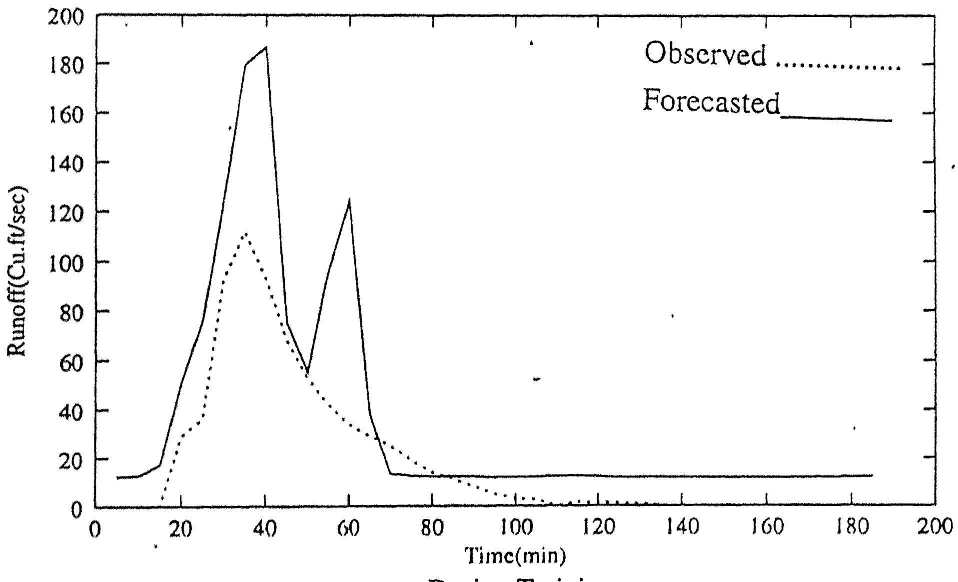


During Testing

Figure 5.3: Observed and Forecasted Runoff from NLMRM 2

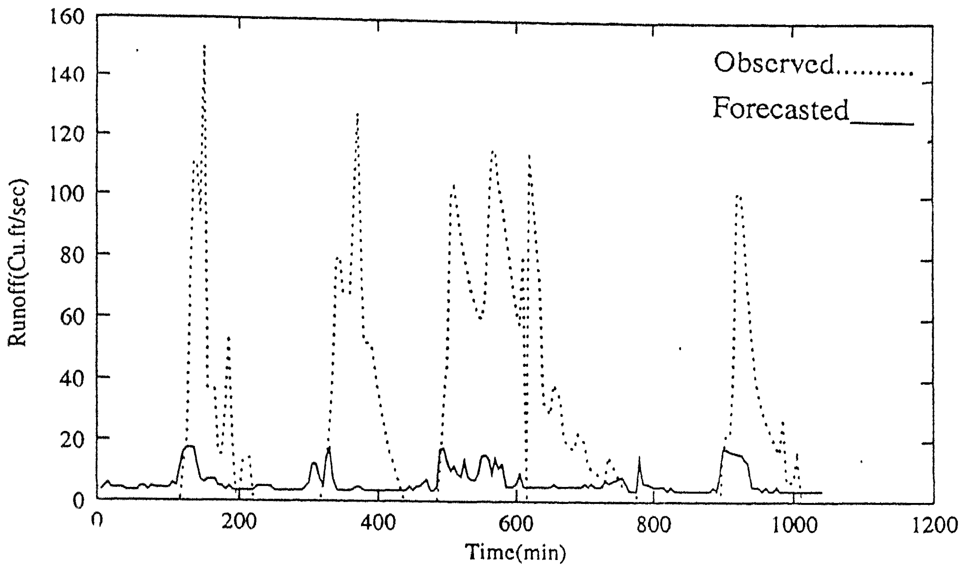


During Testing

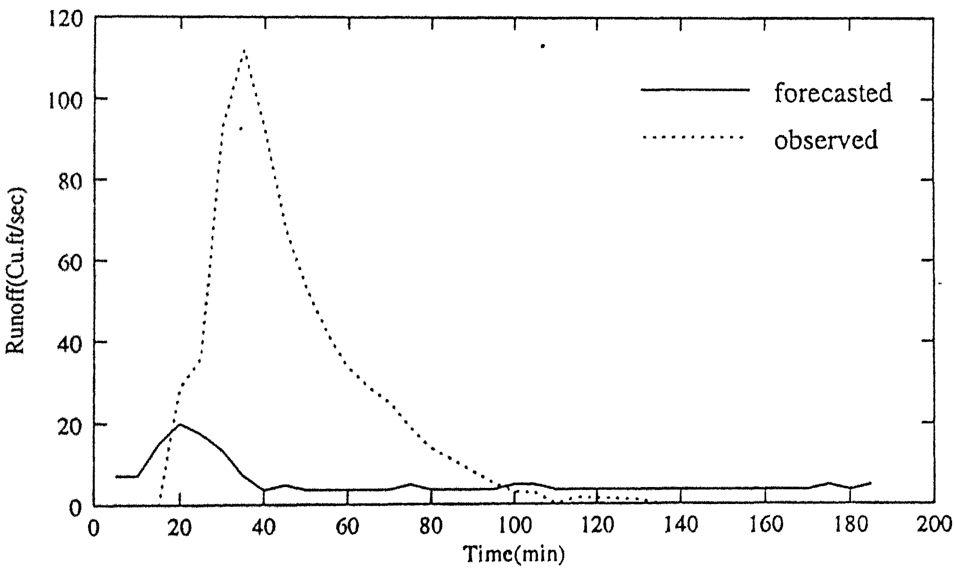


During Training

Figure 5.4: Observed and Forecasted Runoff from NLMRM 3

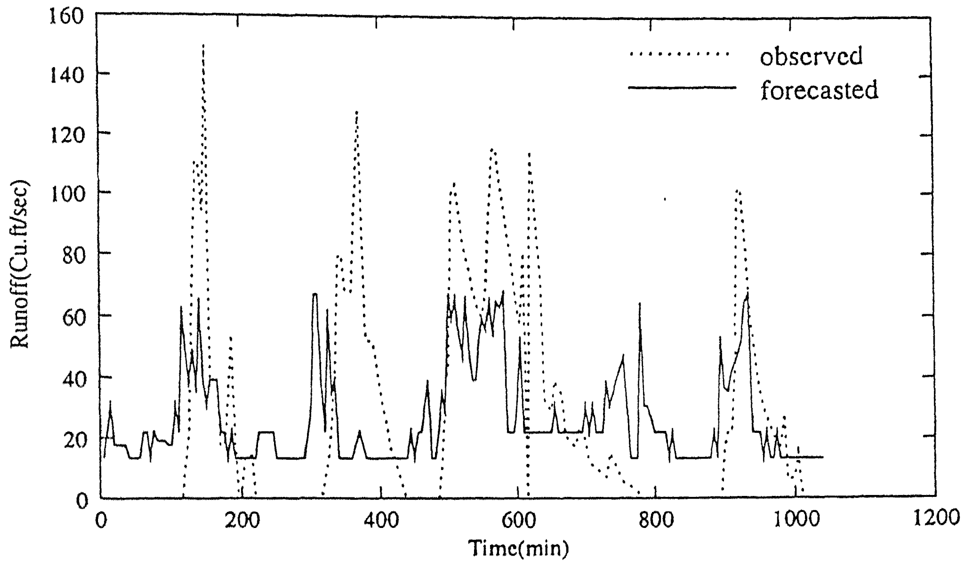


During Training

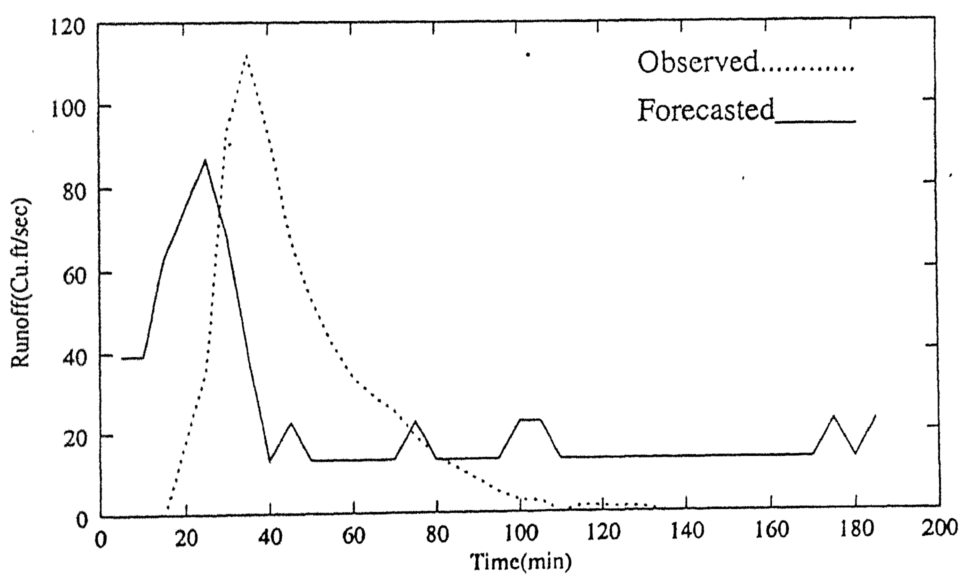


During Testing

Figure 5.5: Observed and Forecasted Runoff from POLY 3

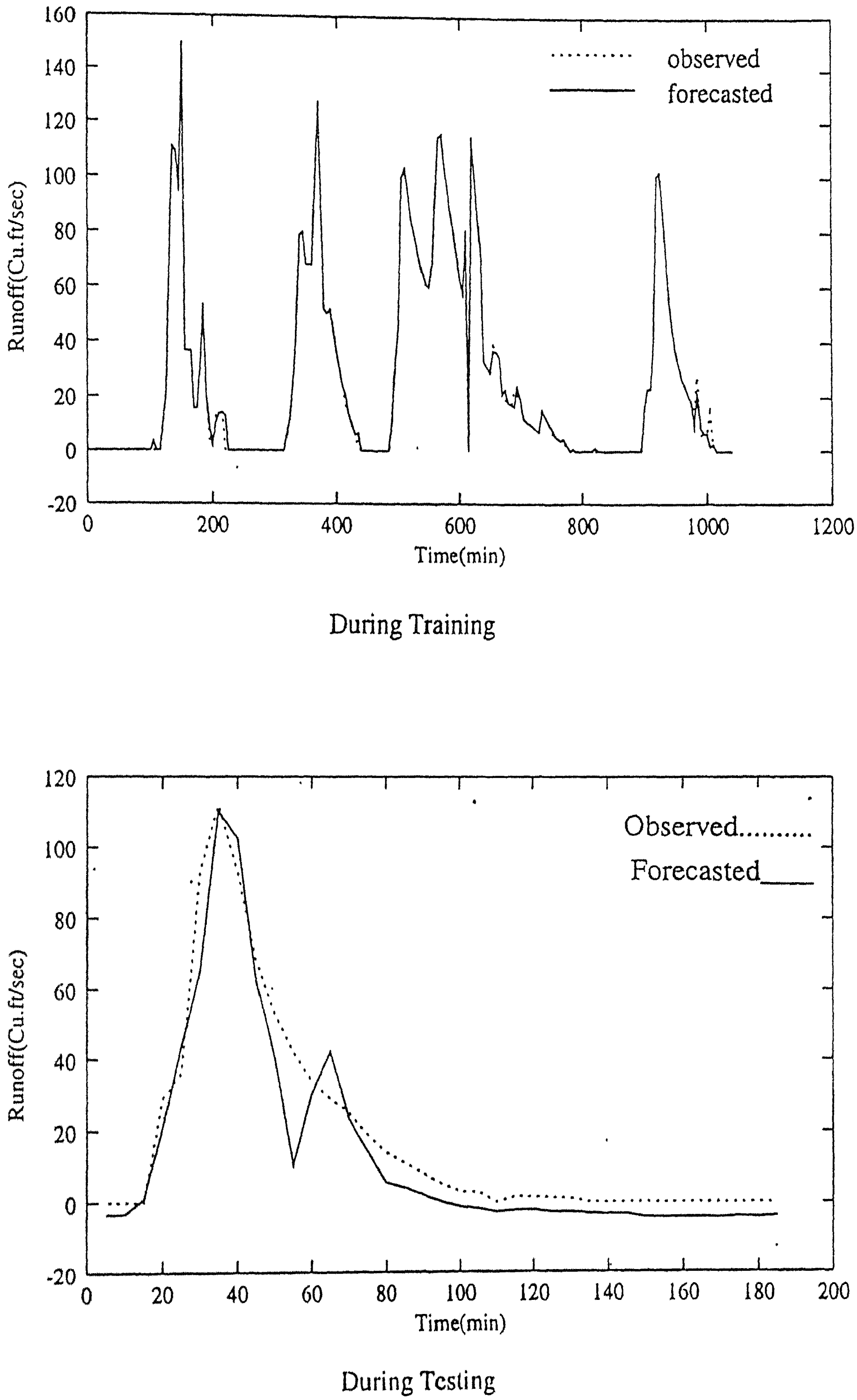


During Training



During Testing

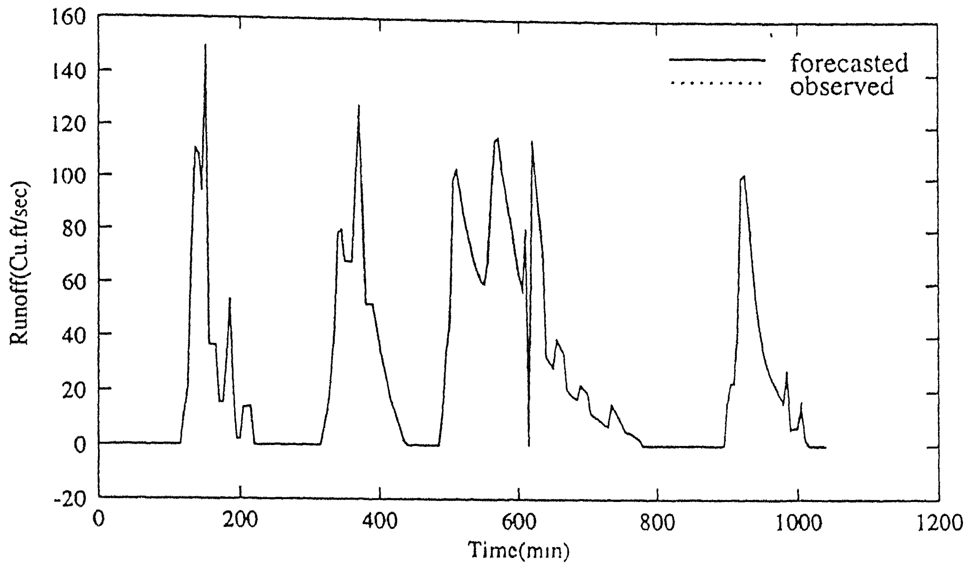
Figure 5.6: Observed and Forecasted Runoff from POLY 6



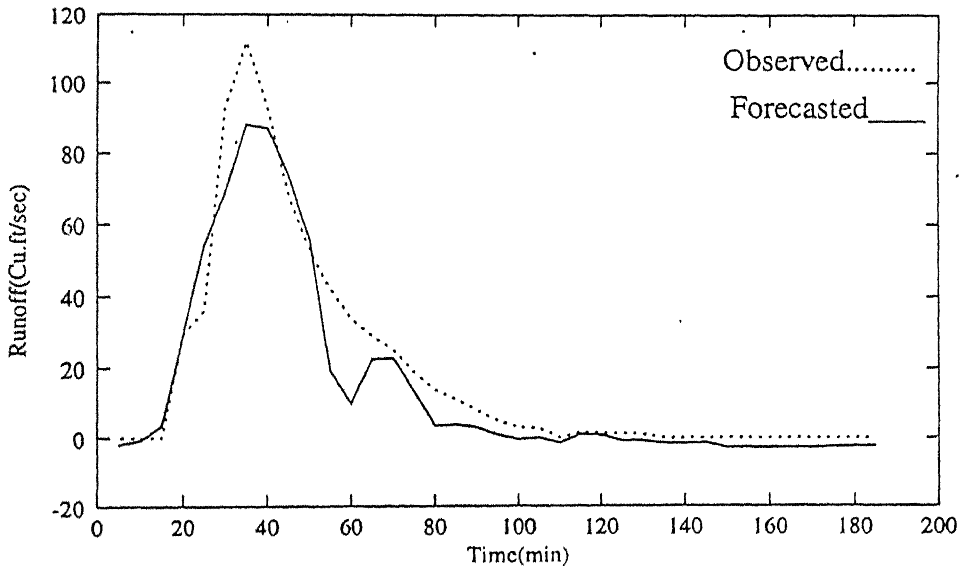
During Training

During Testing

Figure 5.7: Observed and Forecasted Runoff from 10-20-1 ANN Model

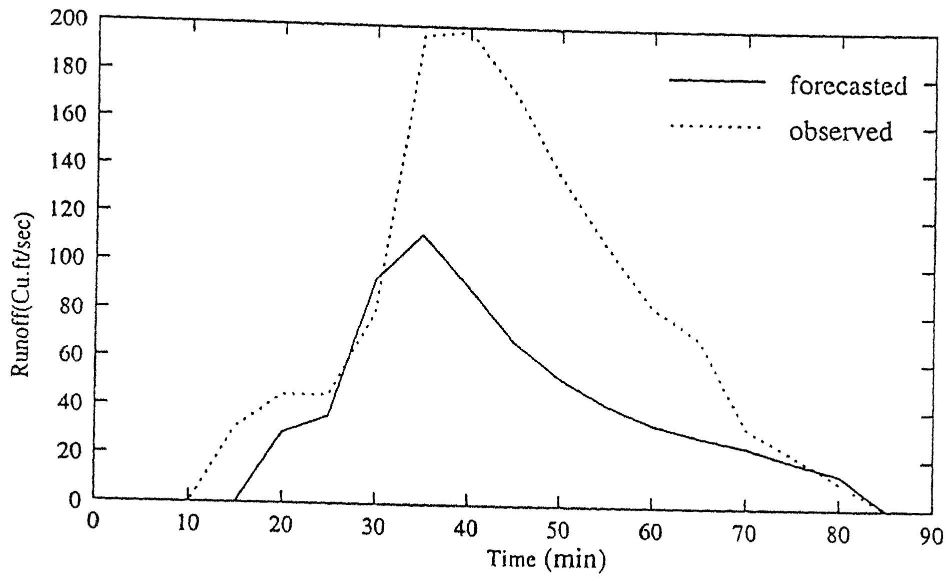


During Training

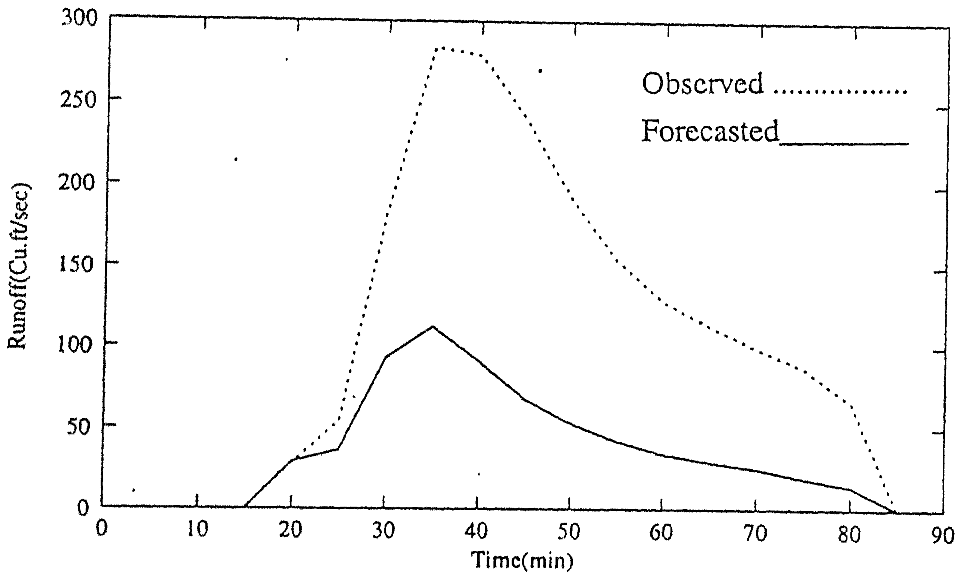


During Testing

Figure 5.8: Observed and Forecasted Runoff from 10-12-14-1 ANN Model



From June 16th Storm



From April 4th Storm

Figure 5.9: Observed and Forecasted Runoff from UH Models

statistics from both ANN models, all five regression models, and the two UH models were averaged. The results of this analysis are presented in table 5.3

As seen from the table 5.3 AARE from ANN models, and regression models is 0.73% and 49.21% during training and 16.59%, and 46.13% during testing respectively. Hence, it can be concluded that, ANN models performed, best in terms of AARE both during training and testing. Correlation coefficients from the ANN models and regression models are 0.9989 and 0.6567 during training and 0.964 and 0.698 respectively during testing respectively. During training TS_0.25 is 94% for 10-12-14-1 ANN Model. The values of RE(Q_p) and RE(t_p)from ANN models are small when comparing with other models. Hence, it can be concluded that, ANN models performed the best in terms of RE(Q_p), RE(t_p) and all threshold statistics during training. However, UH models performed better in terms of TS_1, TS_5 and TS_10 during testing.

5.4 Voting Analysis

A voting analysis was carried-out to select the best model among all the model structures developed in this study. In this voting analysis a model which performs the best in terms of a particular statistic, receives one vote. Total number of votes available are 26. The results of voting analysis are presented in Table 5.5 below. As can be observed, the ANN model 10-12-14-1 recieved 18 out of 26 votes i.e. 70% of the total votes and is deemed to be the best model developed in this study.

Table 5.3: **Average Statistics for Training**

MODEL	AARE	CORR	RE[Qp]	RE[tp]
ANN	0.7395	0.9989	0	0
REGR	49.21	0.6567	14.2	4.2

MODEL	TS_0.25	TS_0.50	TS_0.75	TS_1	TS_5	TS_10	TS_25	TS_50	TS_100
ANN	94.4	94.7	94.7	94.8	96.5	98.1	99.38	99.79	100
REGR	0.36	0.43	0.635	0.835	5.75	11.57	29.64	58.4	91.06

Table 5.4: Average Statistics for Testing

MODEL	AARE	CORR	RE[Qp]	RE[tp]
ANN	16.59	0.964	10.25	1.4
REGR	46.13	0.698	41.41	9.81
UH	86.05	0.564	118.15	11.35

MODEL	TS_0.25	TS_0.5	TS_0.75	TS_1	TS_5	TS_10	TS_25	TS_50	TS_100
ANN	0	0	0	1.35	9.45	25.6	89.18	98.6	1
REGR	0	0	0	0.45	4.21	13.96	32.4	56.9	96.31
UH	0	0	0	7.53	22.522	57.512	53.52	62.5	75.49

Table 5.5: Voting Analysis

MODEL	VOTES
10-20-1 ANN Model	7
10-12-14-1 ANN Model	18
LMR	1
NLMR1	1
POLY3	1
POLY6	0
NLMR2	0
NLMR3	0
UH1	3
UH2	2

Chapter 6

Conclusions

In this study, three types of model structures have been investigated for use in modeling the rainfall-runoff process. The first type of models are ANN models, the second type of models are regression models and the third type of models used the unit hydrograph approach of R-R modeling. Specifically, three ANN models (two three layer 10-20-1 ANN model and another four layer 10-12-14-1 ANN model), six regression models including linear, non-linear and polynomial regression models, and two UH models were developed. The performance of various model structures was evaluated using many statistical parameters. Based on the results obtained in this study, the ANN models have consistently out performed the models that use conventional techniques such as regression and UH method.

ANN technique is relatively new technique which the researchers and scientists have started exploring lately for use in R-R modeling. The technique of ANNs offer excellent tool for modeling and forecasting that can be used to model R-R process. The suitability of ANN models for R-R modeling can also be viewed from the fact that the ANNs are capable of capturing the underlying process of R-R relationship through the least amount of data. In light of the present study, only three storms were available for training purposes but still the ANN models were able to capture the non-linear behavior of the R-R process, better than the other conventional models. However, if the size of data is large then it needs to be explored whether the ANN model would still perform better.

Moreover, back-propagation training algorithm was used to train all the ANNs in this study which has its own limitations. It may be possible to develop a better ANN model for R-R process with other training algorithms such as radial basis functions, genetic-algorithms, unsupervised learning and using fuzzy logic. This aspect also neads to be explored.

Bibliography

[1] Anderson, M.G. and Burt, T.P. (1985), "Hydrological Forecasting", John Wiley and Sons.

[2] Arbib, M.A. (1987), "Brains, Machines and Mathematics", 2nd ed., New York, Springer Verlag.

[3] Beven *et al.* (1980), "SHE: Towards a Methodology for physically Based Distributed Forecasting in Hydrology. In Hydrological Forecasting", IAHS Publication, Vol. 129, pp 133-137

[4] Box G.E. and Jenkins G.M. (1926), "Time Series Analysis : Forecasting and control", Holden-Day, Oakland, California, USA.

[5] Bras R.L. and Rodriguez-I. (1985), "Random Functions and Hydrology", Addition-Wesley, Reading, Massachussets.

[6] Burnash R., Ferrel R., and Mc Guire R., (1973), "A Generalized Streamflow Simulation System, Conceptual modeling for Digital Computers", U.S. National Weather Services, Sacremento, California.

[7] Chang, A.T.C. and Tsang, L. (1992), "A Neural Network Approach to Inversion of Snow Water Equipment from Passive Microwave Measurements", Nord. Hydrol., 23, pp 173-182.

[8] Chang, Y.J., Kavvas, M.L., and Delleur, J.W. (1982), "Stochastic Daily Precipitation Modeling and Daily Stream flow Transfer Process", Technical Report No.146, Water Resources Research Center, Purdue University, West Lafayette, Indiana.

[9] Cho-Chung Yang (1998), PhD student, Department of Civil Engineering National Chiao Tung University, Hsinchu, Taiwan 300, R.O.C.

[10] Churchland, P.S. (1986), "Nemophilosophy: Toward a Unified Science of the Mind/Brain", MIT press, Cambridge, Massachussets.

[11] Churchland, P.S. and Sejnowski, T.J. (1992), "The computational Brain", MIT press, Cambridge, Massachussets.

[12] Crawford, N.H. and Linsley, R.K. (1960), "Computation at a Synthetic Stream flow Record on a Digital Computer", Hydrological Science Bulletin, IAHS publication 51, pp 526-538.

[13] Crawford, N.H. and Linsley, R.K. (1966), "Digital Simulation in Hydrology: Stanford Watershed Model IV", Technical Report 39, Department of Civil Engineering, Stanford University.

[14] Crommelynck *et al.* (1992), "Daily and Hourly Water Consumption Forecasting Tools using Neural Networks", Proceeding of AWWA's Computer Speciality Conference, Nashville, Tennessee, pp 665-676.

[15] Dawdy, D.R., (1983) "A review of Rainfall-Runoff modeling", Proceedings of the Tegucigalpa Hydronath Symposium, September , IAHS Publ. No. 152.

[16] Garrett, J.H., Ranjithan, S. and Eheart, J.W(1992), "Application to Neural Network Ground Water Remediation", In Expert Systems for Civil Engineers: Knowledge Representation, Edited by Allen, R.H., ASCE, NewYork, NewYork, pp 104-143.

[17] Haykin S. (1994), "Neural Networks, A comprehensive Foundation", IEEE Press, Macmillan College Publishing Company.

[18] Hendrickson, J.D., Sorooshian, S., and Brazil, L., (1988), "Comparison of Newton type and Direct Search Algorithms for Calibration of Conceptual Rainfall-Runoff Models", Water Resources Research, 24(5), pp 691-700.

[19] Hydrologic Engineering Center (1981), "HEC-1, Flood Hydrograph Package, Users Manual", U.S. Army Corporation of Engineers, Davis, California.

[20] Jacoby, S.L.S. (1966), "A Mathematical Model for Nonlinear Hydrologic Systems", Journal of Geophysical Research, 71(20).

[21] Johanson, R.C., Imhoff, J.C. and Davis, H.H. (1980), "User Manual for Hydrologic Simulation Program; Fortran (HSPF)", EPA 600/9-80-015, Environmental Protection Agency, Athens, Georgia.

[22] Johnston, P.R. and Pilgria, D.(1976), "Parameter Optimization for Watershed Models", Water Resources Research, 12(3), pp 477-486.

[23] Kuo-lin, H. Gupta, H.V. and Sorooshian S. (1995), "Artificial Neural Network Modeling of the Rainfall-Runoff Process", Water Resources Research, 31(10), pp 2517-2530.

[24] Kuo-lin, H (1998), Department of Hydrology and Water Resources, The University of Arizona. Tueson, AZ85721.

[25] Larimore, W.E. (1981), "Application of the Maximum Likelihood Parameter Identification to Hydrologic Forecasting, Rep. Technical Report No. TR 1480-3, The Anal. Sci. Corp., Reading, Massachussets.

[26] Liurin, F (1994), "Neural Networks in Computer Intelligence", Mc Graw Hill, Inc.

[27] Livine, M. (1985), "Man and Machin Visison", New York, Mc Graw Hill, Inc.

[28] Mary, D. (1982), "Vision", Freeman, New York.

[29] Mason, J.C., Price R.K. and Tem'me A. (1997), "A Neural Network Model of Rainfall-Runoff using Radial Basis Functions", Journal of Hydralic Research, 34(4), pp 537-548.

[30] Mc Cuen, R.H. and Snyder, W.M. (1986), "Hydrological Modeling: Statistical Methods and Applications", Prentice Hall Publications.

[31] Morris, E.M (1980), "Forecasting Flood Flows in Grassy and Forested Basins Using a Deterministic Distributed Mathematical Model", Publication No.129, International Association of Scientific Hydrology, Willing ford, England, pp 247-255.

[32] O'Connel P.E.(1974), "Stochastic Modeling of Long Term Persistence in Stream Flow Sequence", Ph.D. Thesis, Imperial College, University of London, England.

[33] Park *et. al.* (1991), "Electrical Load Forecasting using an Artificial Neural Network", IEEE Transactions on Power Systems, Vol.6, No.2, pp 442-449.

[34] Pegram, G.G.S. (1980), "A Continuous Stream Flow Model", Journal of Hydrology, 47, pp 65-89.

[35] Pickup, G. (1977), "Testing of Efficiencies of Algorithms and Strategies for Automatic Calibration of Rainfall-Runoff Models", Hydrogeol. Science Bulletin, 22(2), pp 257-274.

[36] Ponce, V.M. (1989), "Engineering Hydrology : Principles and Practice", by Prentice Hall Publications.

[37] Rehak, D.R., Thewalt, C.R. and Doo, L.B. (1989), "Neural Network Approach in Structural Mechanics Computations", The Proceedings of the Sessions related to Comuter Utilization at Structural Congress'89, ASCE, NewYork, pp 168-176.

[38] Salas, J.D., Delleus, J.W., Yerjerich, V. and Lane, W.L. (1980), "Applied Modeling in Hydrologic Time Series", Water Resources Publications, Littleton, Colorado.

[39] Sharda, R. and Patil, R.B. (1990), "Connectionist Approach to Time Series Prediction: An Empirical Test", Technical report No. 90-26: 11-90, College of Business Administration, Oklahoma State University, Oklahoma.

[40] Sherman, L.K. (1932), "Stream flow from Rainfall by a Unit Hydrograph Method", Engineering News Record, Vol 108, pp 501-505.

[41] Sorooshian, S. (1978), "Considerations of Stochastic Properties in Parameter Estimation of Hydrologic Rainfall-Runoff Models", Ph.D. Dissertation, University of California, Los Angels.

[42] Sorooshian, S. and Gupta V.K. (1983), "Automatic Calibration of Conceptual Rainfall Runoff models: The Question of Parameter Observability and Uniqueness", Water Resouces Research, 19(1), pp 260-268.

[43] Sorooshian S., Duan Q. and Gupta V.K. (1993), "Calibration of Rainfall-Runoff Models: Application to Global Optimization to the Sacramento Soil Moisture Accounting Model", Water Resources Research, 29(4), pp 1185-1194.

[44] Sugawara, M. (1961), "An Analysis of Runoff Structure about Several Japanese Rivers", Japanese Journal of Geophysics, 2.

[45] Treiber B. and Plate E.J. (1977), "A Stochastic Model for Simulation of Daily Flows", AWRA's Hydrological Science Bulletin, 22(1).

[46] U.S. Standard Supplements (1966), National Aeronautical and Space Administration.

[47] Wasserman, P.D. (1989), "Neural Computing Theory and Practice", New York: Van Nostrand Reinhold.

[48] Zurada, J.M.(1997), "Introduction to Artificial Neural Systems", Jaico Publishing House.

APPENDICES

APPENDIX A

RAINFALL-RUNOFF DATA FOR SALADO CREEK AT BITTERS
STATION NO. 08178690
FOR APRIL 4, MAY 6-7, JUNE 16, AND AUG 18

RAINFALL (in)	RUNOFF (Cu.ft/sec)
0.000000	0.000000
0.010000	0.000000
0.020000	0.000000
0.005000	0.000000
0.005000	0.000000
0.005000	0.000000
0.005000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.010000	0.000000
0.010000	0.000000
0.000000	0.000000
0.010000	0.000000
0.006600	0.000000
0.006600	0.000000
0.006600	0.000000
0.005000	0.000000
0.005000	0.000000
0.020000	0.000000
0.010000	0.000000
0.070000	0.000000
0.170000	11.07384
0.270000	20.93980
0.320000	71.07435
0.240000	110.9405
0.080000	108.9271
0.030000	94.02774
0.020000	150.0012
0.030000	36.64456
0.030000	36.64456
0.030000	36.64456
0.010000	15.50357
0.010000	15.50357
0.000000	31.00695
0.010000	53.96014
0.000000	19.93310
0.000000	2.013390
0.000000	2.013392
0.000000	13.89274
0.000000	13.89274
0.000000	14.09411
0.000000	0.000000
0.010000	0.000000

0.010000	0.000000
0.010000	0.000000
0.010000	0.000000
0.010000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.010000	0.000000
0.020000	0.000000
0.090000	0.000000
0.090000	0.000000
0.030000	0.000000
0.010000	6.442929
0.130000	13.08742
0.230000	24.96658
0.030000	42.08098
0.000000	78.92692
0.000000	79.93362
0.000000	68.05426
0.000000	68.05426
0.000000	68.05426
0.005000	98.05452
0.010000	128.0547
0.005000	88.18875
0.000000	51.94675
0.000000	51.94675
0.000000	51.94675
0.000000	44.49694
0.000000	36.04061
0.000000	30.00025
0.000000	23.95989
0.000000	16.91283
0.000000	13.89274
0.000000	10.06714
0.000000	6.040363
0.000000	1.610826
0.000000	0.000000
0.010000	0.000000
0.000000	0.000000
0.010000	0.000000
0.010000	0.000000
0.020000	0.000000
0.030000	0.000000
0.000000	0.000000
0.000000	0.000000
0.010000	0.000000
0.230000	10.87265
0.410000	33.02034
0.100000	45.10107
0.060000	99.86653
0.080000	103.0881

0.050000	95.03443
0.040000	85.97379
0.110000	79.93362
0.040000	75.10132
0.030000	68.85958
0.030000	65.03418
0.050000	61.00720
0.140000	60.00051
0.150000	68.05426
0.120000	94.02774
0.050000	114.9673
0.120000	115.9740
0.070000	103.8935
0.090000	97.04782
0.010000	87.98738
0.010000	81.14168
0.010000	71.87986
0.020000	63.02060
0.050000	57.98712
0.010000	80.94032
0.010000	0.000000
0.010000	115.1687
0.010000	101.0746
0.010000	85.57122
0.010000	71.87986
0.010000	33.02034
0.010000	31.00695
0.010000	28.99356
0.020000	39.06071
0.010000	36.64456
0.010000	34.42979
0.010000	20.93980
0.010000	19.12760
0.010000	17.91952
0.010000	17.11420
0.010000	22.34925
0.010000	20.93980
0.020000	19.53035
0.010000	12.08072
0.020000	10.87265
0.010000	10.06714
0.010000	9.060450
0.010000	7.852381
0.030000	7.047060
0.020000	15.10081
0.025000	12.68467
0.030000	10.46990
0.035000	7.651004
0.040000	5.033668
0.020000	4.832291
0.000000	4.026972
0.000000	3.020089
0.000000	2.013392
0.120000	0.000000
0.020000	0.000000
0.020000	0.000000
0.015000	0.000000

0.010000	0.000000
0.010000	0.000000
0.010000	0.000000
0.010000	0.000000
0.000000	0.000000
0.010000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.010000	0.000000
0.000000	0.000000
0.050000	0.000000
0.260000	15.90613
0.250000	23.15457
0.190000	22.95319
0.180000	41.07410
0.170000	100.8734
0.160000	102.0814
0.120000	88.99406
0.100000	70.87316
0.010000	55.97354
0.010000	44.89969
0.010000	36.04061
0.000000	31.00695
0.010000	26.98016
0.000000	23.95989
0.000000	20.93980
0.010000	18.12090
0.000000	15.90613
0.000000	27.98686
0.000000	6.040363
0.000000	6.644306
0.000000	6.644306
0.000000	14.89944
0.000000	2.416140
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.030000	0.000000
0.030000	0.000000
0.140000	0.000000
0.520000	28.99356
0.360000	36.04061
0.110000	93.02085
0.030000	111.9472
0.000000	93.02085

0.010000	68.05426
0.000000	52.95345
0.000000	42.08098
0.000000	33.82584
0.000000	28.99356
0.000000	25.16796
0.010000	18.92641
0.000000	13.89274
0.000000	11.07384
0.000000	8.053755
0.000000	5.033668
0.010000	3.020089
0.010000	2.818899
0.000000	0.000000
0.000000	1.610826
0.000000	1.409449
0.000000	1.208073
0.000000	1.006696
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.000000	0.000000
0.010000	0.000000
0.000000	0.000000
0.010000	0.000000

APPENDIX B

NASA DATA FOR TEMPERATURE FOR DIFFERENT ALTITUDE

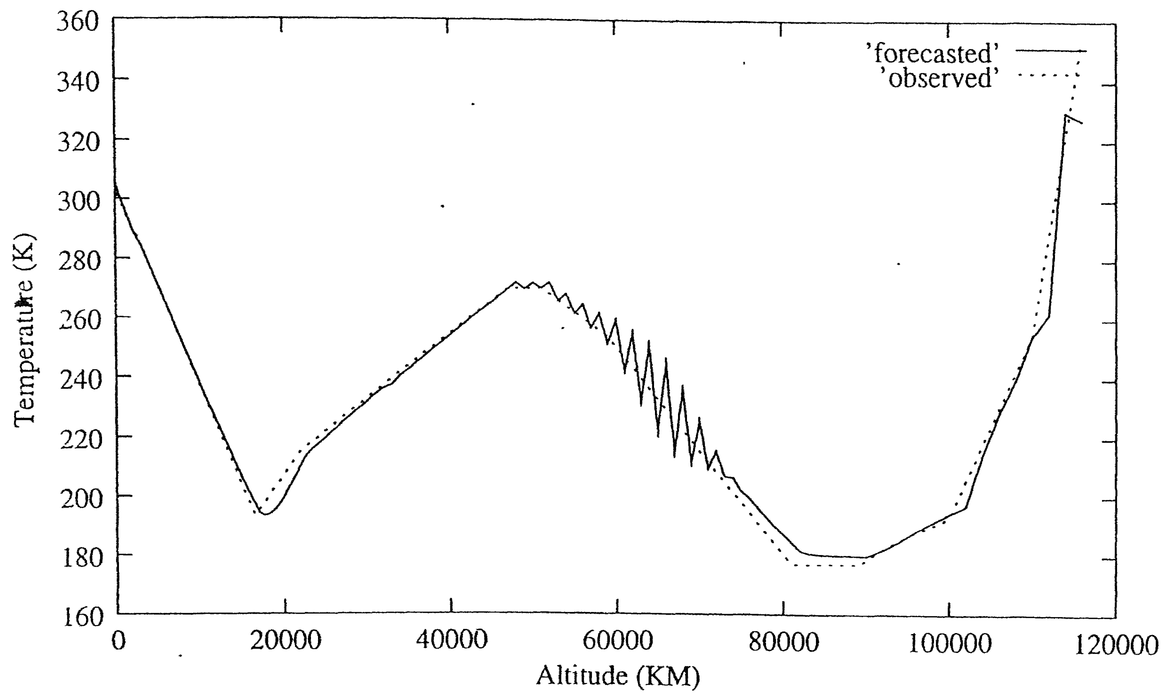
ALTITUDE (KM)	TEMPERATURE (K)
0.000000	302.589996
250.000000	300.910004
500.000000	299.239990
750.000000	297.570007
1000.000000	295.890015
1250.000000	294.250000
1500.000000	292.609985
1750.000000	290.980011
2000.000000	289.339996
2250.000000	287.720001
2500.000000	287.739990
2750.000000	286.010010
3000.000000	284.279999
3250.000000	282.549988
3500.000000	280.820007
3750.000000	279.089996
4000.000000	277.359985
4250.000000	275.660004
4500.000000	273.950012
4750.000000	272.239990
5000.000000	270.540009
5250.000000	268.829987
5500.000000	267.119995
5750.000000	265.420013
6000.000000	263.709991
6250.000000	262.019989
6500.000000	260.320007
6750.000000	258.630005
7000.000000	256.940002
7250.000000	255.250000
7500.000000	253.559998
7750.000000	251.860001
8000.000000	250.169998
8250.000000	248.490005
8500.000000	246.889999
8750.000000	245.130005
9000.000000	243.440002
9250.000000	241.759995
9500.000000	240.080002
9750.000000	238.399994
10000.000000	236.720001
10250.000000	235.039993
10500.000000	233.360001
10750.000000	231.679993
11000.000000	230.000000
11500.000000	226.649994
12000.000000	223.300003

12500.000000	219.949997
13000.000000	216.600006
13500.000000	213.250000
14000.000000	209.899994
14500.000000	206.550003
15000.000000	203.199997
15500.000000	199.850006
16000.000000	196.500000
16500.000000	193.149994
17000.000000	195.149994
17500.000000	197.149994
18000.000000	199.149994
18500.000000	201.149994
19000.000000	203.149994
19500.000000	205.149994
20000.000000	207.149994
20500.000000	209.149994
21000.000000	211.149994
21500.000000	213.149994
22000.000000	215.149994
22500.000000	216.250000
23000.000000	217.350006
23500.000000	218.449997
24000.000000	219.550003
24500.000000	220.649994
25000.000000	221.750000
25500.000000	222.850006
26000.000000	223.949997
26500.000000	225.050003
27000.000000	226.149994
27500.000000	227.250000
28000.000000	228.350006
28500.000000	229.449997
29000.000000	230.449997
29500.000000	231.649994
30000.000000	232.750000
30500.000000	233.850006
31000.000000	234.949997
31500.000000	236.050003
32000.000000	237.149994
33000.000000	239.350006
34000.000000	241.550003
35000.000000	243.750000
36000.000000	245.949997
37000.000000	248.149994
38000.000000	250.350006
39000.000000	252.550003
40000.000000	254.750000
41000.000000	256.950012
42000.000000	259.149994
43000.000000	261.350006
44000.000000	263.549988
45000.000000	265.750000
46000.000000	267.950012
47000.000000	270.149994
48000.000000	270.149994
49000.000000	270.149994

50000.000000	270.149994
51000.000000	270.149994
52000.000000	268.149994
53000.000000	266.149994
54000.000000	264.149994
55000.000000	262.149994
56000.000000	260.149994
57000.000000	258.149994
58000.000000	256.149994
59000.000000	254.149994
60000.000000	250.649994
61000.000000	247.149994
62000.000000	243.649994
63000.000000	240.149994
64000.000000	236.649994
65000.000000	233.149994
66000.000000	229.649994
67000.000000	226.149994
68000.000000	222.649994
69000.000000	219.149994
70000.000000	215.649994
71000.000000	212.149994
72000.000000	208.649994
73000.000000	205.149994
74000.000000	201.649994
75000.000000	198.149994
76000.000000	194.649994
77000.000000	191.149994
78000.000000	187.649994
79000.000000	184.149994
80000.000000	180.630005
81000.000000	177.119995
82000.000000	177.110001
83000.000000	177.100006
84000.000000	177.080002
85000.000000	177.070007
86000.000000	177.059998
87000.000000	177.050003
88000.000000	177.029999
89000.000000	176.919998
90000.000000	178.669998
92000.000000	182.160004
94000.000000	185.320007
96000.000000	187.869995
98000.000000	190.419998
100000.000000	192.979996
102000.000000	205.460007
104000.000000	217.839996
106000.000000	230.130005
108000.000000	242.320007
110000.000000	254.470001
112000.000000	288.190002
114000.000000	321.670013
116000.000000	354.950012

RESULT FOR ALTITUDE AND TEMPERATURE DATA OF NASA

AARE	TS0025	TS0050	TS0075	TS_1	TS_5	TS_10	TS_25	TS_50	TS_75	TS_100	CORR
0.051466	0.0028	0.0443	0.2341	0.3038	0.3481	0.6898	0.7784	0.9873	1.0000	1.0000	0.9892



SIMULATION OF ALTITUDE AND TEMPERATURE DATA OF NASA

APPENDIX D

RESULTS FROM 9-N-1 ANN MODELS FOR R-R MODELING

MODEL	AARE	TS_1	TS_5	TS_10	TS_25	TS_50	TS_100
9_2_1	0.641666	0.000000	0.057143	0.089796	0.216327	0.432653	0.734694
9_3_1	0.470122	0.004082	0.028571	0.069388	0.236735	0.481633	0.975510
9_4_1	0.380773	0.008163	0.036735	0.093878	0.232653	0.804082	1.000000
9_5_1	0.342398	0.000000	0.032653	0.085714	0.297959	0.816327	1.000000
9_6_1	0.320444	0.000000	0.024490	0.077551	0.436735	0.816327	1.000000
9_7_1	0.302965	0.004082	0.036735	0.081633	0.538776	0.812245	1.000000
9_8_1	0.287681	0.008163	0.040816	0.142857	0.571429	0.812245	1.000000
9_9_1	0.278660	0.004082	0.044898	0.200000	0.571429	0.820408	1.000000
9_10_1	0.267359	0.016327	0.106122	0.289796	0.571429	0.816327	1.000000
9_11_1	0.258098	0.073469	0.191837	0.367347	0.575510	0.828571	1.000000
9_12_1	0.254384	0.061224	0.248980	0.383673	0.571429	0.828571	1.000000
9_13_1	0.253102	0.069388	0.269388	0.383673	0.567347	0.828571	1.000000
9_14_1	0.252965	0.048980	0.314286	0.383673	0.559184	0.828571	1.000000
9_15_1	0.253383	0.057143	0.257143	0.404082	0.559184	0.836735	1.000000
9_16_1	0.254435	0.065306	0.269388	0.412245	0.559184	0.828571	1.000000
9_17_1	0.255819	0.048980	0.212245	0.408163	0.555102	0.828571	1.000000
9_18_1	0.257288	0.048980	0.220408	0.400000	0.555102	0.828571	1.000000
9_19_1	0.258739	0.061224	0.204082	0.408163	0.555102	0.816327	1.000000
9_20_1	0.260315	0.065306	0.191837	0.404082	0.551020	0.816327	1.000000
9_21_1	0.261857	0.048980	0.195918	0.412245	0.551020	0.816327	1.000000

APPENDIX E

RESULT FOR 9-N-N-1 NETWORK FOR DIFFERENT NEURONS
TRAINING AND TESTING ON ALL DATA POINTS

MODEL	AARE	TS0025	TS0050	TS0075	TS_1	TS_5	TS_10	TS_50	TS_100
9_8_8_1	0.095638	0.604082	0.604082	0.608163	0.689796	0.714286	0.832653	0.951020	1.000000
9_9_8_1	0.008212	0.967347	0.967347	0.967347	0.971429	0.979592	0.983673	0.995918	1.000000
9_9_10_1	0.155051	0.697959	0.697959	0.702041	0.718367	0.738775	0.763265	0.848980	0.975510
9_10_10_1	0.000731	0.975510	0.979592	0.979592	0.995918	1.000000	1.000000	1.000000	1.000000
9_11_12_1	0.035671	0.865306	0.873469	0.877551	0.893878	0.906122	0.934694	0.975510	1.000000
9_12_12_1	0.004173	0.963265	0.963265	0.971429	0.975510	0.987755	0.991837	1.000000	1.000000
9_12_13_1	0.002791	0.938776	0.946939	0.959184	0.979592	0.987755	1.000000	1.000000	1.000000
9_11_14_1	0.009771	0.918367	0.918367	0.918367	0.959184	0.967347	0.987755	1.000000	1.000000
9_12_14_1	0.000327	0.987755	0.991837	0.991837	1.000000	1.000000	1.000000	1.000000	1.000000
9_18_19_1	0.011408	0.930612	0.930612	0.934694	0.963265	0.971429	0.979592	0.991837	1.000000
9_18_20_1	0.007113	0.922449	0.926531	0.926531	0.983673	0.983673	0.983673	1.000000	1.000000
9_19_20_1	0.012225	0.930612	0.930612	0.934694	0.967347	0.971429	0.979592	0.987755	1.000000

APPENDIX F

RESULTS FOR 10-N-1 NETWORK

MODEL	AARE	SD	TS-0.25	TS_0.50	TS_1	TS_5	TS_10	TS_25	TS_50	TS_100	CORR
10_15_1	0.015875	0.889796	0.893878	0.897959	0.897959	0.910204	0.955102	0.979592	0.995918	1.000000	0.99867
10_16_1	0.015102	0.881633	0.889796	0.893878	0.902041	0.926531	0.951020	0.987755	0.995918	1.000000	0.998781
10_17_1	0.019691	0.865306	0.869388	0.869388	0.869388	0.906122	0.934694	0.975510	0.995918	1.000000	0.998329
10_18_1	0.016229	0.885714	0.897959	0.897959	0.902041	0.934694	0.955102	0.979592	0.991837	1.000000	0.998574
10_19_1	0.017157	0.877551	0.877551	0.881633	0.885714	0.914286	0.930612	0.983673	0.995918	1.000000	0.998660
10_20_1	0.014093	0.889796	0.893878	0.897959	0.897959	0.930612	0.963265	0.987755	0.995918	1.000000	0.998952
10_21_1	0.018227	0.869388	0.881633	0.881633	0.881633	0.906122	0.942857	0.979592	0.995918	1.000000	0.998277
10_22_1	0.024207	0.840816	0.848980	0.848980	0.853061	0.889796	0.922449	0.971429	0.991837	1.000000	0.997498

RESULT FOR 10-N1-N2-1 NETWORK

MODEL	AARE		TS_025	TS_050	TS_075	TS_1	TS_5	TS_10	TS_25	TS_100	corr
9_10_10_1	0.001402	0.955102	0.959184	0.967347	0.975510	1.000000	1.000000	1.000000	1.000000	1.000000	0.9999
9_10_11_1	0.019534	0.885714	0.885714	0.885714	0.885714	0.918367	0.942857	0.967347	1.000000	1.000000	0.9807
9_12_11_1	0.004896	0.963265	0.967347	0.967347	0.967347	0.983673	0.991837	0.991837	1.000000	1.000000	0.9999
9_12_12_1	0.001734	0.971429	0.971429	0.975510	0.975510	0.991837	1.000000	1.000000	1.000000	1.000000	0.9999
9_12_14_1	0.000777	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.9999
9_12_15_1	0.007958	0.922449	0.930612	0.930612	0.934694	0.955102	0.963265	0.991837	1.000000	1.000000	0.9996